A New Level of Trusted Cloud Computing - Virtualized Reconfigurable Resources in a Security-First Architecture

Paul R. Genssler,[1] Oliver Knodel,[2] Rainer G. Spallek[3]

Abstract: The cloud business is growing year by year, but there are still a lot of security concerns. The users of cloud systems, from a single individual to big companies, have to trust multiple parties who could all interfere and disrupt the provided service or compromise data privacy and security. New software exploits are being found every day, therefore we propose a secured reconfigurable hardware architecture to house sensitive algorithms and data. Using vetted cryptographic techniques and the slight inaccuracy of todays chip-manufacturing to generate unique keys only available to the hardware itself, we are able to create a trusted compute space in a remote cloud system. To achieve typical cloud characteristics like resource pooling and rapid elasticity, the reconfigurable silicon is virtualized while preserving the high security standard.

Keywords: Reconfigurable Computing, Cloud, Security, Virtualization

1 Motivation

The seamless user experience in today's information age is powered by an endless farm of servers. They are the foundations for an ever growing market of cloud computing [Gar17]. The cloud offers virtually infinite on demand processing power and scales fast to absorb a sudden spike in user requests. However, the server's high power consumption limits further growth of datacenters hence providers are exploring how to improve the performance through alternative architectures like GPUs or FPGAs. GPUs are a common solution nowadays, while reconfigurable hardware accelerators are slow to catch up but gain more momentum every year. In 2014 Mircosoft tested FPGAs to speed up their bing search [Pu14] and rolled out a second version to their data centers worldwide in 2016 [Ca16]. At the end of the same year Amazon announced their new EC2 F1 Instances [Ama16] making Xilinx UltraScale Plus FPGAs available to customers.

But regardless of whether the client uses CPUs, GPUs or FPGAs, they always have to trust the cloud provider to keep their data and algorithms secure and private. To reduce the chain of trust and keep the client's data safe but also to reduce power consumption and improve performance a new reconfigurable security-first architecture is needed.

[1] Technische Universität Dresden paul.genssler@tu-dresden.de
[2] Technische Universität Dresden oliver.knodel@tu-dresden.de
[3] Technische Universität Dresden rainer.spallek@tu-dresden.de

doi:10.18420/in2017_48

The following section 2 will outline previous work, after that the threat and trust model is described, section 4 introduces the new architecture followed by a security analysis in section 5. The paper is concluded in section 6.

2 Related Work

This section outlines literature from different research areas fundamental to secure reconfigurable hardware in the cloud. First, security concerns in cloud computing are summarized followed by approaches to control and minimize the threats and problems. Subsequently, virtualization of FPGAs is introduced and security concerns mentioned. The section concludes by showing different proposals for remote but secure FPGA designs.

2.1 Security Concerns in Cloud Computing

A cloud is primarily based on a software system to manage the clients. Like almost every software it is vulnerable to exploits and weak configurations opening security holes. However, there are more risks the client has to accept when using remote resources. A very detailed overview was given by Fernandes ([Fe14]) which will be summarized hereafter. It is not known what happens to the stored data: if it is duplicated, altered in any way, thoroughly isolation from other client's data, reliably overwritten before reuse or if there will be any downtimes. Virtualization introduces a whole lot of new security issues like virtual machine (VM) image theft or code injection. These issues persist even when the servers go offline, the security of old images degrades due to discovery of new vulnerabilities. The virtual machine manager (VMM) is a single point of failure as well as a worthwhile target to attack multiple VMs at once. A virtualized network allows packet sniffing and spoofing, leads to unstable network characteristics and can reduce the effectiveness of traditional security methods. VMs themselves can be the target of man in the middle or side-channel attacks or malware injection. Additionally a malicious administrator is a threat the client cannot control as well as identity management, authentication and authorization procedures.

2.2 Approaches on Cloud Security

A common tool to secure a standard CPU-based machine is a Trusted Platform Module (TPM) [Tru]. It provides features like authenticated boot sequence or cryptographic keys. It does not prevent modifications of a running program or data extraction, hence it cannot protect the whole system sufficiently. An FPGA based TPM module was proposed by Eisenbarth in [Ei07], but it is geared towards a processor supported single user application and requires an initial setup by a trusted third party. The sensitive internal state of the module cannot be transfered, making migration between nodes very difficult, preventing a flexible cloud.

Another approach for secure computing in a cloud is the use of full homomorphic encryption schemes described by Gentry ([Ge09]). They execute special algorithms on encrypted data - without decrypting it at any point. It is a very elegant solution, the user only has to take care of the key management. However, Moore concluded in their survey ([Mo14]) that there are still a lot of open problems and research to be done. For example could Lee attack parts the algorithm because of an incorrect choice of parameters ([Le11]). They summarized further that the performance of the used algorithms has to be drastically improved, homomorphic encryption is far too slow to be used in a productive system.

2.3 Preparing FPGAs for the Cloud

Following Moore's Law, the number of transistors doubles approximately every two years, FPGAs are growing in size with every new generation. But not every design can make use of the huge amount of available resources using only a portion of the chip. To increase the utilization, the logic can be virtualized to allow multiple different designs on the same device. This approach was used by El-Araby in [EAGEG08] to virtually increase the number of reconfigurable accelerators available to a CPUs although there was only a single physical device. Byma ([By14]) divided the logic resources into multiple regions which can be allocated like standard VMs using the open source cloud software OpenStack. An extended interface was proposed in [FVS15] by Fahmy, allowing not only prioritized communication to the host, but also provided an interface between the single virtual FPGAs. The focus of Knodel in [KS15] is not only on virtualization of the FPGA allowing flexible partitioning but also on different service models to satisfy various customer demands.

The before mentioned proposals focused on enabling FPGAs in the cloud, however, they neglected major security concerns. Although software offers a larger attack surface than hardware, due to their complex multi tasking and resource sharing, FPGAs and their configuration are still vulnerable to certain attacks and multiple security concerns arise.
First of all, the system designer cannot be sure that the chip is free of hardware Trojans, either the FPGA vendor or the foundries could have added them. However, Agrawal investigated in [Ag07] how the vendor can detect hidden backdoors using methods from cryptanalysis like power and temperature profiling to create a unique fingerprint for the IC. To generate such a fingerprint, only a few chips have to be invasively tested if they comply with the vendor's specifications. In 2012 a backdoor leaking security keys among others was detected in a military grade Actel/Mircosemi FPGA ([SW12]).
Thompson challenges the basic trust in software in [Th84]. The EDA tools themselves could extract information or manipulate the developers designs. This basic problem directly affects all approaches on security and is not within the scope of this paper. Meanwhile a strong reputation of the EDA tools developer might ease the concerns of system designers and clients.
Operating FPGAs or other hardware under direct control of the clients is fairly secure, e.g. can firewalls prevent any data leakage. Yet, many FPGAs are deployed in the field

and the system designer looses direct physical access to secure the system. It has to withstand attempts of reverse engineering, cloning or overbuilding, among others. Leading manufactures are aware of such threats and attacks and provide tools to secure the valuable intellectual property (IP) [Tr07]. However, the system has its shortcomings. Trimberger outlines possible attacks in [TM14]. For example, the decrypted design could be intercepted while it configures the FPGA. Side-channel attacks like power, timing and electromagnetic emanation analysis can be used to weaken the protection from encryption. Timing based ([Bh09]), optical ([SA03]) or EM ([SH07]) fault injections can break the security of the chip as well. But again, there are effective defenses in place to protect the chip which were summarized in [ZQ14] by Zhang.

If the engineer has access to the devices before they are deployed, necessary security measures can be taken. However, in a cloud environment where the Infrastructe as a Service (IaaS) provider cannot be trusted, it is impossible for the client to establish any trust. Therefore third party entities, often called trusted authority (TA), are employed in different proposals to deploy a secured initial configuration. Drimer ([DK09]) and Devic ([DTB10]) describe both a protocol for secure remote updates which could also be used in the cloud context. They require additional Non-Volatile-Memory (NVM) and the first configuration is done in a trusted environment. Kepa proposed in [Ke08] a few enhancements to the FPGA fabric to allow for a secure controller with a strong chain of trust to be implemented. A TA certifies the public key generated by the controller. They mediate access to the internal configuration port to prevent a malicious configuration from altering their controller. But the user space would have to be free of any routing resources used by the static design of the controller for this approach to work. Another single user framework was proposed by Eguro ([EV12]). With the example of medical data they described how a TA would configure FPGAs with symmetric keys before they are delivered to the datacenter. Afterwards, clients send their design to the TA for encryption. It than can only by used by the trustworthy preconfigured FPGA. They point out the difficulties of a secure multi user approach because of route-through wires to hard macros or I/O pins which have to be protected from an adversary.

But employing a TA just transfers the trust problem from one party to another, the number of involved and trusted participants does not change from the clients perspective.

3 Security Assumptions

Based on the literature analysis of the previous section, this paper proposes a new architecture. It provides flexible virtualization of reconfigurable hardware and allows a remote but secure deployment in the cloud. There, according to Mell multiple providers have to be considered ([Me11]). At the basic level is the owner of the datacenter, the IaaS provider. On top of that is the Platform as a Service (PaaS) provider, controlling operating systems and configurations to virtualize the basic resources enabling a consumer to deploy applications. The Software as a Service (SaaS) provider uses the platform to offer services like web-mail to various clients.

3.1 Threat Model

The proposed system is considered being under attack and in an unsafe environment after it has been shipped from the chip vendor (CV) to the board manufacturer. Therefore, the board itself is treated as hostile and no unencrypted data can leave the chip, unless explicitly sent by the user through a dedicated channel. Only standard cryptographic algorithms like AES and RSA are used in this proposal, therefore all encrypted data is assumed to be secure without access to the key. Furthermore, the chip is hardened against physical attacks which would yield the cryptographic keys easily. Costlier attempts are not efficient enough to scale well and are therefore not considered a threat to the proposed system in general. The security architecture was designed to protect the user's data and algorithms. Hence, slow downs, denial of service attacks or even physical destruction are not evaluated.

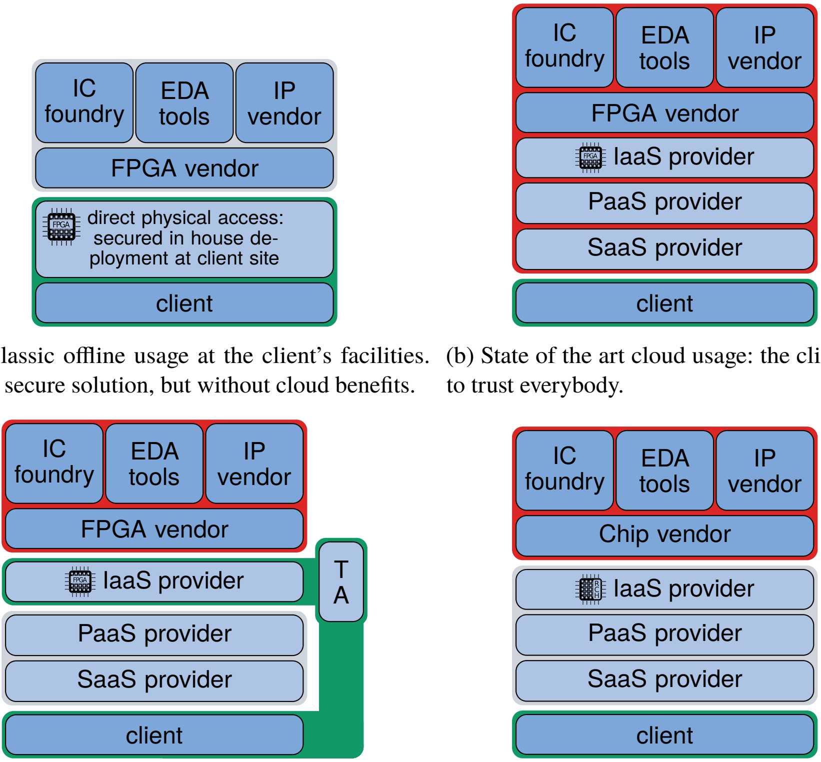
3.2 Trust Model

The user of a any remote system never has total control over it contrary to classic in-house deployment shown in Figure 1a. There, not even data leaking hardware Trojans are a concern because any network access can be physically prevented or fully controlled by firewalls. However, the trend towards cloud computing moves the devices out of client's reach and into datacenters controlled by the IaaS provider. Other service provider base their business model on top and the client is compelled to trust them all. Only the local workplace can be considered secure (Figure 1b).

To reduce the required trust, several approaches were proposed and outlined in section 2.3. However, they either only virtualize the FPGA without sufficient save guards or secure it but fail to meet the cloud characteristics of resource pooling i.e. do not allow multiple user. Those proposals rely on a TA to establish trust, to secure the first deployment or to sign keys associated with a FPGA (Figure 1c). This intermediate step is necessary because the chips and vendors do not offer better features geared to the needs of cloud security. Therefore, the proposed architecture is the next logical step uniting virtualization and security into a single chip whilst no TA or a secure environment for the first setup is needed (Figure 1d). Instead, the initial configuration is part of the chip, untouchable neither by any other party involved in the its deployment nor through malicious user. This creates a trustworthy compute space for the clients.

4 The Secure Architecture

The proposed new architecture is a specialized FPGA for secure cloud computing in an untrusted environment. However, trough virtualization it becomes an elastic cloud resource



(a) Classic offline usage at the client's facilities. Most secure solution, but without cloud benefits.

(b) State of the art cloud usage: the client has to trust everybody.

(c) Based loosely on proposal from [EV12], using a trusted authority (TA) which the client and the Iaas Provider have to trust. Other providers can only access encrypted data.

(d) The proposed new architecture does not rely on any third party and also excludes the IaaS provider.

Fig. 1: An overview of typical offline (a) and cloud (b) usage models. (c) includes a third party TA. The proposed new architecture shown in (d) minimizes the required trust.

like storage or CPU cores. This section describes the modules of the architecture and how they are engineered to work together to ensure security.

4.1 System Design

An abstract view of the chip layout is given in Figure 2 showing the division in reconfigurable regions (RRs) and a static part. The basic modules are fixed and cannot be tampered with by the users, the board manufacturer or the cloud infrastructure provider. This includes endpoints for various high speed interfaces, e.g. PCIe for device-host communication, Ethernet for direct network access, a board-to-board connection or other protocols. The

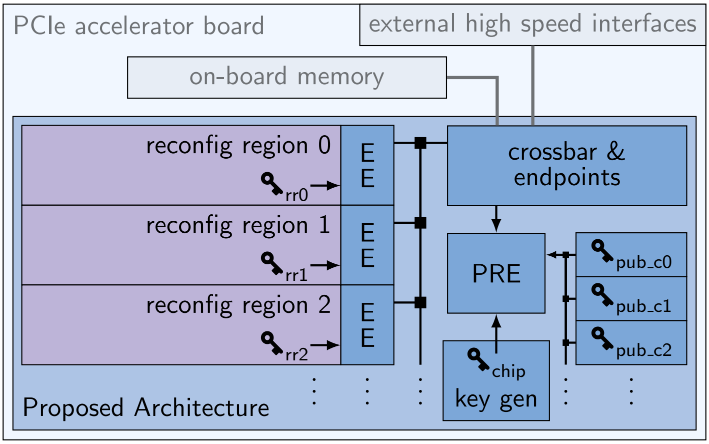


Fig. 2: The chip is located on a typical accelerator board with fast memory and various connections (e.g. PCIe, QPI or Ethernet). The crossbar is an endpoint for those links and manages the communication of the reconfigurable region (RR), which is always secured through the encryption engines (EEs) supplied with a symmetric key directly from the RR. A RR is reconfigured through the partial reconfiguration engine (PRE) which gets a private key from the key generator. The public keys of the clients are written by the host into the pub_cX registers to verify their digital signature on hardware level.

implementation of those endpoints includes a simple round-robin flow control to prevent a malicious user from flooding it and blocking of other customers. An advanced version would include a more sophisticated approach to conform to different service level agreements dependent on the user's requirements. The crossbar handles requests to the on-board memory which is subdivided into as many partitions as there are RRs. It blocks all request which would violate this direct association, although this is only an extra security layer.

Because anyone with physical access to the board could read out the memory contents, each byte of data leaving the RR is encrypted by the encryption engine (EE) regardless if it goes to the memory or through the high speed interfaces. The EEs are high speed Advanced Encryption Standard (AES) cores using a key directly supplied from within each of the RRs. The user specified symmetric key is embedded in the configuration data, hereafter also named bitstream. To prevent it from being stolen the bitstream header is encrypted using the public key of the chip.

This asymmetric key pair is generated by the silicon itself using the key generator (*key gen* in Figure 2). This module is implemented using a physically unclonable function (PUF) to create an individual key pair for every single chip. There are multiple proposals (e.g. [SD07], [PD11]) on how to build a reliable key generator using such an element. Only the public key is accessible from the outside and is published by the chip vendor through a public key infrastructure (PKI). The private key is solely readable by the partial reconfiguration engine (PRE), the module to alter the configuration of the RR. It decrypts the incoming bitstream header, which contains an user specified symmetric key to decrypt the rest of the bitstream. Starting with asymmetric encryption is mandatory. An implementation using only symmetric algorithms requires the distribution of the secret key to every user who could in turn decrypt bitstreams of other users. However, symmetric encryption is significantly faster and should be used to improve reconfiguration performance.

Aside from the static part, most area of the chip is reconfigurable like current SRAM-based FPGAs. This area is virtualized in multiple homogeneous RRs therefore allowing the combination of an arbitrary number to a larger region assignable to a single user. This allows for bigger and more complex designs and avoids different chip versions with different RR sizes and numbers. It is the basis for a flexible and scalable system required in the cloud.

4.2 Hypervisor

A node in the cloud system contains at least one board with the proposed chip and is virtualized by the PaaS provider, who uses a VMM or hypervisor to serve every user a separate VM. This is a common approach, but the hypervisor has to perform additional tasks. Alongside with the RR allocation request the user sends a public key which is forwarded to the PRE, the hardware side of the hypervisor. There the key is written into a register associated with that region. The user signs the checksum of the bitstream with the corresponding private key and appends it. The PRE uses the public key stored in the register to verify it and to prevent tampering, cloning or other malicious attempts to alter the bitstream.

To prevent hijacking of other users' space and interfering with their task or even extracting unencrypted data, the bitstream consists of single frames. Each has an address which is used by the PRE to filter out any frames not located in the currently reconfigured region. If multiple regions are required, the public key is written into each of the registers and the configuration process is repeated for every region. After a successful and verified reconfiguration, the PRE notifies the crossbar of the changes and it flushes any buffers and clears the allocated on-board memory. The unused EEs are disabled and the data is routed through a single port.

4.3 User Design Flow

The design flow form the user's point of view does not change a lot compared to traditional FPGA development chains. Initially an interface is available to model the high speed connections and the on-board memory. The EE is transparent to the user. After the design was synthesized the tools show how many regions are required and partition the bitstream accordingly. It is encrypted with a user generated symmetric key which itself will be secured with the public key from the target chip.

5 Security Analysis

The security of the proposed system is based on the key generator and the unique private key it generates. A commercial design is therefore expected to implement additional state of the art defenses against fault, power or other kind of attacks to protect this key from extraction ([Fo10]). Hence, the client has to trust in the ability of the CV to produce effective but also Trojan free hardware. However, it is in the interest of the company to deliver the most secure devices. It takes a long time to build a strong reputation, but only one misstep to ruin it. Also Anderson points out the "scatter gun" nature of reconfigurable devices in [ANY08]. The client may rely on the fact that the vendor does not know which design will be executed using the chip, therefore cannot directly target a single customer. Implementing some sort of discovery mechanism would increase the chance of Trojans being detected even further. A TA on the other hand knows its clients and could select targets more specifically. At this point a basic level of trust in well known manufactures has to be assumed otherwise the general usage of connected electronic devices has to be questioned.

6 Conclusion and Future Work

This paper described a novel architecture to provide a secure compute space in the cloud. Based on a asymmetric cryptographic key generated directly in hardware and unable to leave the device, the system can establish a high level of trust for the client. Other than previous proposals outlined in section 2.3, this novel architecture does not rely on a secured first boot or a third party trusted authority. It protects the client's algorithms and data from all cloud participants and even the chip vendor using standard cryptographic processes. Through virtualization of the reconfigurable resources within the device, resource pooling as well as fast and flexible scalability are possible – key characteristics of a successful cloud.

Right now, the client has to encrypt parts of the design with the specific public key of each chip. To allow quick and easy scaling to multiple chips this process might be moved to the cloud. A bootstrap design could be installed to handle the encryption. The actual design's bitstream would be uploaded to it and stored securely through it.
Another problem is defragmentation of the reconfigurable space. Moving a few smaller tasks from multiple nodes to a single node makes room for larger requests increasing the overall system efficiency. Migration of a design state was shown by Knodel ([KGS17]), but they broke the security boundary by using the host's CPU to extracted the state from the unencrypted bitstream. It has to be investigated how a transparent relocation between regions or general pausing and extracting can be done whilst keeping the design secure.
The immediate next step is a prototype implementation of the proposed architecture using a FPGA. This will allow us to determine resource usage and overhead of such a model. However, based on previous implementations of similar frameworks we do not expect any significant bottlenecks considering the vastly improved security this architecture provides.

References

[Ag07] Agrawal, D.; Baktir, S.; Karakoyunlu, D.; Rohatgi, P.; Sunar, B.: Trojan Detection using IC Fingerprinting. In: 2007 IEEE Symposium on Security and Privacy (SP '07). pp. 296–310, May 2007.

[Ama16] Amazon. Amazon EC2 F1 Instances, November 2016. https://aws.amazon.com/ec2/instance-types/f1/.

[ANY08] Anderson, Matthew Smith; North, CJG; Yiu, Kenneth K: Towards Countering the Rise of the Silicon Trojan. Technical report, 2008.

[Bh09] Bhasin, S.; Selmane, N.; Guilley, S.; Danger, J. L.: Security evaluation of different AES implementations against practical setup time violation attacks in FPGAs. In: 2009 IEEE International Workshop on Hardware-Oriented Security and Trust. pp. 15–21, July 2009.

[By14] Byma, S.; Steffan, J. G.; Bannazadeh, H.; Garcia, A. L.; Chow, P.: FPGAs in the Cloud: Booting Virtualized Hardware Accelerators with OpenStack. In: 2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines. pp. 109–116, May 2014.

[Ca16] Caulfield, Adrian M; Chung, Eric S; Putnam, Andrew; Angepat, Hari; Fowers, Jeremy; Haselman, Michael; Heil, Stephen; Humphrey, Matt; Kaur, Puneet; Kim, Joo-Young et al.: A cloud-scale acceleration architecture. In: Microarchitecture (MICRO), 2016 49th Annual IEEE/ACM International Symposium on. IEEE, pp. 1–13, 2016.

[DK09] Drimer, Saar; Kuhn, Markus G: A protocol for secure remote updates of FPGA configurations. In: International Workshop on Applied Reconfigurable Computing. Springer, pp. 50–61, 2009.

[DTB10] Devic, Florian; Torres, Lionel; Badrignans, Benoit: Secure protocol implementation for remote bitstream update preventing replay attacks on FPGA. In: Field Programmable Logic and Applications (FPL), 2010 International Conference on. IEEE, pp. 179–182, 2010.

[EAGEG08] El-Araby, Esam; Gonzalez, Ivan; El-Ghazawi, Tarek: Virtualizing and sharing reconfigurable resources in High-Performance Reconfigurable Computing systems. In: High-Performance Reconfigurable Computing Technology and Applications, 2008. HPRCTA 2008. Second International Workshop on. IEEE, pp. 1–8, 2008.

[Ei07] Eisenbarth, Thomas; Güneysu, Tim; Paar, Christof; Sadeghi, Ahmad-Reza; Schellekens, Dries; Wolf, Marko: Reconfigurable trusted computing in hardware. In: Proceedings of the 2007 ACM workshop on Scalable trusted computing. ACM, pp. 15–20, 2007.

[EV12] Eguro, Ken; Venkatesan, Ramarathnam: FPGAs for trusted cloud computinga. In: Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on. IEEE, pp. 63–70, 2012.

[Fe14] Fernandes, Diogo AB; Soares, Liliana FB; Gomes, João V; Freire, Mário M; Inácio, Pedro RM: Security issues in cloud environments: a survey. International Journal of Information Security, 13(2):113–170, 2014.

[Fo10] Fournaris, Apostolos P: Fault and simple power attack resistant RSA using Montgomery modular multiplication. In: Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on. IEEE, pp. 1875–1878, 2010.

[FVS15] Fahmy, S. A.; Vipin, K.; Shreejith, S.: Virtualized FPGA Accelerators for Efficient Cloud Computing. In: 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom). pp. 430–435, Nov 2015.

[Gar17] Gartner. Gartner Says Worldwide Public Cloud Services Market to Grow 18 Percent in 2017, February 2017. http://www.gartner.com/newsroom/id/3616417.

[Ge09] Gentry, Craig: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st annual ACM symposium on Symposium on theory of computing - STOC '09. ACM Press, pp. 169–169, 2009.

[Ke08] Kepa, Krzysztof; Morgan, Fearghal; Kosciuszkiewicz, Krzysztof; Surmacz, Tomasz: Serecon: A secure dynamic partial reconfiguration controller. In: Symposium on VLSI, 2008. ISVLSI'08. IEEE Computer Society Annual. IEEE, pp. 292–297, 2008.

[KGS17] Knodel, Oliver; Genssler, Paul R.; Spallek, Rainer G.: Migration of Long-running Tasks Between Reconfigurable Resources Using Virtualization. SIGARCH Comput. Archit. News, 44(4):56–61, January 2017.

[KS15] Knodel, O.; Spallek, R. G.: Computing Framework for Dynamic Integration of Reconfigurable Resources in a Cloud. In: 2015 Euromicro Conference on Digital System Design. pp. 337–344, Aug 2015.

[Le11] Lee, Moon Sung: On the sparse subset sum problem from Gentry-Halevi's implementation of fully homomorphic encryption. IACR Cryptology ePrint Archive, 2011:567, 2011.

[Me11] Mell, Peter; Grance, Tim et al.: The NIST definition of cloud computing. 2011.

[Mo14] Moore, Ciara; O'Neill, Maire; O'Sullivan, Elizabeth; Doroz, Yarkin; Sunar, Berk: Practical homomorphic encryption: A survey. In: Circuits and Systems (ISCAS), 2014 IEEE International Symposium on. IEEE, pp. 2792–2795, 2014.

[PD11] Paral, Zdenek; Devadas, Srinivas: Reliable and efficient PUF-based key generation using pattern matching. In: Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on. IEEE, pp. 128–133, 2011.

[Pu14] Putnam, A.; Caulfield, A. M.; Chung, E. S.; Chiou, D.; Constantinides, K.; Demme, J.; Esmaeilzadeh, H.; Fowers, J.; Gopal, G. P.; Gray, J.; Haselman, M.; Hauck, S.; Heil, S.; Hormati, A.; Kim, J. Y.; Lanka, S.; Larus, J.; Peterson, E.; Pope, S.; Smith, A.; Thong, J.; Xiao, P. Y.; Burger, D.: A reconfigurable fabric for accelerating large-scale datacenter services. In: 2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA). pp. 13–24, June 2014.

[SA03] Skorobogatov, Sergei P.; Anderson, Ross J.: Optical Fault Induction Attacks. In (Kaliski, Burton S.; Koç, çetin K.; Paar, Christof, eds): Cryptographic Hardware and Embedded Systems - CHES 2002: 4th International Workshop Redwood Shores, CA, USA, August 13–15, 2002 Revised Papers. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 2–12, 2003.

[SD07] Suh, G Edward; Devadas, Srinivas: Physical unclonable functions for device authentication and secret key generation. In: Proceedings of the 44th annual Design Automation Conference. ACM, pp. 9–14, 2007.

[SH07] Schmidt, Jörn-Marc; Hutter, Michael: Optical and em fault-attacks on crt-based rsa: Concrete results. 2007.

[SW12] Skorobogatov, Sergei; Woods, Christopher: Breakthrough Silicon Scanning Discovers Backdoor in Military Chip. In (Prouff, Emmanuel; Schaumont, Patrick, eds): Cryptographic Hardware and Embedded Systems – CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 23–40, 2012.

[Th84] Thompson, Ken: Reflections on trusting trust. Communications of the ACM, 27(8):761–763, 1984.

[TM14] Trimberger, Steve; Moore, Jason: FPGA security: From features to capabilities to trusted systems. In: Proceedings of the 51st Annual Design Automation Conference. ACM, pp. 1–4, 2014.

[Tr07] Trimberger, Steve: Trusted design in FPGAs. In: Proceedings of the 44th annual Design Automation Conference. ACM, pp. 5–8, 2007.

[Tru] Trusted Computing Group. TPM Main Specification Level 2. https://trustedcomputinggroup.org/tpm-main-specification.

[ZQ14] Zhang, J.; Qu, G.: A survey on security and trust of FPGA-based systems. In: 2014 International Conference on Field-Programmable Technology (FPT). pp. 147–152, Dec 2014.