# A survey on security and trust of FPGA-based systems

**Article** · April 2015

DOI: 10.1109/FPT.2014.7082768



**2 authors**, including:

Jiliang Zhang
Northeastern University (Shenyang, China)
**18** PUBLICATIONS **107** CITATIONS

SEE PROFILE

# A Survey on Security and Trust of FPGA-based Systems

Jiliang Zhang
College of Information Science and Engineering
Hunan University
Changsha, 410082 China
Email: hnu.zjl@gmail.com

Gang Qu
Department of Electrical and Computer Engineering
University of Maryland
College Park, MD 20742 USA
Email: gangqu@umd.edu

***Abstract*—This survey reviews the security and trust issues related to FPGA-based systems from the market perspective. For each party involved in FPGA supply and demand, we show the security and trust problems they need to be aware of and the solutions that are available.**

## I. Introduction

The battle between ASIC (application specific integrated circuit) and FPGA (field programmable gate array) has been around for more than three decades and it will not end in the near future. With the advances in semiconductor technologies and the increased design complexity, the high capacity and highly flexible FPGA has become a design platform for a large variety of systems, which we refer to as FPGA-based systems. There are many excellent tutorials, surveys and books on all aspects of the FPGA-based system design. In this section, we provide a short description of these works and define the scope of this survey to distinguish our work from theirs.

The fundamental of FPGA-based system design can be found in a handful of textbooks such as [1]. A survey on the architecture and design challenges of modern commercial FPGA is given in [2]. As the first comprehensive survey on FPGA design automation, [3] elaborates both the basics and new advances in all major phases in FPGA design flow. These and many other similar works focus on FPGA concepts and design issues of FPGA-based systems, security and trust are not discussed.

The 2004 paper by Wollinger, Guajardo, and Paar [4] is the first survey on FPGA security covering the following three topics: the advantages of using FPGA for cryptographic applications; the security vulnerabilities and existing attacks to FPGAs; and the available countermeasures against these attacks. Drimer's 2008 report [5] gives a more in-depth discussion and classification of attacks to FPGA-based systems. It also provides more modern issues such as trust, adversary classification, and security metrics. Majzoobi, Koushanfar, and Potkonjak [6] present a complementary view of the problem in their 2011 book chapter entitled "FPGA-Oriented Security". In addition to a detailed analysis of vulnerabilities in both FPGA synthesis flow and FPGA-based systems, they discuss FPGA security primitives by the examples of physical unclonable functions (PUF) and true random number generators (TRNG) and top FPGA security challenges. Another 2011 book [7], "Security Trends for FPGAs: From Secured to Secure Reconfigurable Systems", consists of contributions from the following five topics: security FPGA analysis, side channel attack, countermeasures against physical attacks, FPGA TRNG, and embedded systems security for FPGA.

Besides the above survey works, there are many papers and articles focused on problems related to FPGA security and trust. Here we list several more works that cover more general FPGA security problems and leave the rest to the later sections when we survey specific topics. In 2002, Kean presents a commercial model that allows FPGA intellectual property (IP) core vendors to sell their IP on a pay-per-use basis [8]. In 2007, Trimberger highlights the importance of trusted design in FPGAs [9] while Gu, Qu, and Zhou propose a novel framework to build trust at early system design stages [10]. In a 2008 article by Huffmire et al. [11], encryption, avionics, and computer vision are used as examples to show the security management problem in FPGA-based embedded system where the authors promote a holistic approach that includes life-cycle management. In 2014, Trimberger and Moore describe the security features used in present-day FPGAs [12]. A 2014 book chapter by Durvaux et al. [13] presents the FPGA implementation of AES, the performance evaluation and resistance against side-channel and fault attacks. They also explain recent topics such as FPGA bitstream security, watermarking, and PUF. Recently, Zhang et al. [14] present the concept of reconfigurable binding that reconfigures the traditional static PUF and the locking mechanism to prevent FPGA bitstream from replay attacks.

Unlike a comprehensive tutorial, or a review of a certain topic, on FPGA-based system design, we consider the FPGA supply and demand model (see Fig. 1), the parties involved in FPGA market and the security and trust challenges facing these parties. In this model, on the supply side, FPGA vendors will build their FPGA families through semiconductor foundry and make them available on the market; on the demand side, end users will request system developer to design an FPGA-based system to implement the desired application. It is our goal to analyze the potential damages that can hurt each of the parties, and to make them aware of the exiting solutions to protect themselves.

In Section II, we will describe this model, the role of each party in the model, and the security and trust vulnerabilities in the interactions among the parties. In Section III, we survey the solutions that have been developed to resist the potential attacks. Section IV concludes the paper.

978-1-4799-6245-7/14/$31.00 ©2014 IEEE

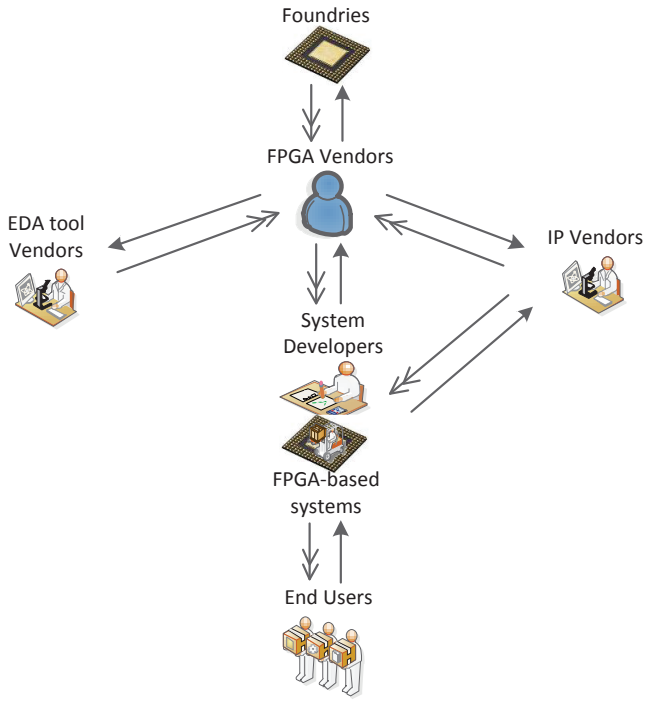


Fig. 1. The supply and demand flows in the FPGA-based system market. "→": service requesting; "→→": service providing.

## II. Vulnerabilities and Attacks

### A. FPGA Market Model

We consider the following major parties in the FPGA-based systems market:

- FPGA vendors: these are the companies (like Xilinx and Altera) that design FPGA architecture and build FPGA chips. These chips normally come with some built-in functional units, memory blocks, and other IP components that the FPGA vendors believe the customers will need.
- Foundries: these are the semiconductor manufacturers (like TSMC, UMC, Globalfoundries, IBM) that fill fabricate the FPGA chips for the FPGA vendors.
- FPGA-based system developers: these are the companies that create commercial products on FPGA chips. The product is normally in the form of configuration bitstream file for a given family FPGA chips.
- FPGA-based system end users: these are companies or individuals who obtain the FPGA-based systems (like network routers, TV set-top boxes) from the system developer and use these systems.
- IP core vendors: these are the companies that develop IP cores (like memory blocks, DSP cores) for specific applications, not necessary for FPGA-based systems.
- EDA tool vendors: these are the companies that develop software tools to facilitate the design of large scale integrated circuits, including FPGA chips.

### B. Supply and Demand Flow

The FPGA market supply flow starts from FPGA vendors, who will introduce new families of FPGA chips to maintain their respective competitive edges in the market. The FPGA vendors will work with IP core vendors to include new IPs on their chips in order to better meet the requirements from emerging applications of end users. FPGA vendors also need to work with EDA tool vendors to enhance the design toolkits associated with their FPGA chips. Finally, FPGA vendors need semiconductor foundries to fabricate the FPGA chips.

This flow and the interactions among different parties are shown on the top half of Fig. 1, where an arrow with single arrowhead indicates the request of a service and an arrow with double arrowhead indicates providing the service. For example, FPGA vendor asks an IP core vendor to develop a specific functional unit to be placed on the FPGA chips (as an IP) and the IP core vendor delivers.

Similarly, the bottom half of Fig. 1 shows the demand flow of the FPGA-based system market, which will be initiated by end users. Users, except those who have their own hardware system developing team, give their system specification and design requirements to system developers with design expertise. The system developing process normally involves the acquisition of third party IPs and the use of licensed EDA tools to generate the FPGA configuration bitstream file that defines the FPGA-based system based on users requirements. Most systems are built on the existing FPGA chips in the market. When there is the need for specialized FPGA chip with dedicated IP components, customized FPGA chips can be fabricated, normally through the FPGA vendors.

### C. Security and Trust Vulnerabilities

We now analyze the vulnerabilities at each interaction between a pair of parties in the above FPGA-based systems. Same notion (e.g. hardware Trojan or IP protection) may be used for similar vulnerabilities under different scenario. The general guideline is that on the sending side of an arrow, the party that sends out the request will have concerns related to the security of the information or product he sends out; on the receiving end of an arrow, the party needs to verify whether the received service or product is trusted.

FPGA vendors and foundries: (a) **overbuilding**: a dishonest foundry may fabricate more FPGA chips than the vendors have requested and sell them at a lower price to system developers; (b) **hardware Trojan**: during the fabrication process, unwanted functionalities, known as hardware Trojan, may be embedded in the FPGA chips; (c) **information leaking**: FPGA vendors confidential data needed for the fabrication of the FPGA chips may be mishandled or leaked to parties (e.g. another competing FPGA vendor) that should not have access to such data.

FPGA vendors (or system developers) and IP core vendors (or EDA tool vendors): (b) **hardware Trojan**: FPGA vendors (and system developers) need to ensure that the acquired IPs from the core vendors do not possess malicious Trojans; (c) **information leaking**: malicious codes in EDA tools may collect valuable data about FPGA chip and/or the system to be built on the FPGA chip; (d) **IP protection**: IP core vendors

and EDA tool vendors want that their IPs and tools are used and royalty are paid properly. (e) **reverse engineering**: IP core vendors expect their IPs cannot be reverse engineered by untrusted parities.

FPGA vendors and system developers: (b) **hardware Trojan** and (c) **information leaking**: there is no guarantee that FPGA chip does not contain hardware Trojan and the EDA tools do not have malicious codes. However, system developers have limited options and have to trust the FPGA chips and the associated design tools to design the systems.

System developers and end users: both parties involved in this interaction need to protect their respective IPs. System developers face several security challenges unique to FPGA-based system. (e) **reverse engineering**: System developers expect their products cannot be reverse engineered by untrusted parities. (f) **cloning**: the FPGA configuration bitstream is obtained by eavesdropping or from the volatile SRAM and used to configure FPGA chips; (g) **side channel attacks**: when bitstream file is encrypted as offered by most FPGA vendors, side channel attacks can reveal the keys stored in the FPGA chips and makes the bitstream file unprotected; (h) **FPGA replay attack**: a dishonest user downgrades an FPGA-based system to the previous version of FPGA chips with known vulnerabilities and explores such vulnerabilities.

## III. The State-of-the-art Defenses

From the perspective of the FPGA security and trust, overbuilding, cloning, hardware Trojans, reverse engineering, side-channel and replay, are considered to be the common security vulnerabilities of volatile FPGAs. For each attack, we elaborate the state-of-the-art defenses.

### A. Overbuilding

An untrusted foundry may overbuild FPGAs and sell them at a lower price to system developers. The main countermeasure is the hardware metering [15]. Hardware metering is a set of tools, methodologies, and protocols that enable passive or active controlling of the number of produced ICs. In the passive hardware metering [16][17], inherent uniqueness derived from an unclonable manufacturing variability is leveraged to uniquely identify each IC. In the active metering [18][19], the original FSM was modified to lock the function of the design and/or IP which can be unlocked only by the designer and/or IP vendors. The detailed introduction about the hardware metering, please refer to the recent surveys by Koushanfar [15][20].

### B. Hardware Trojan

With widespread outsourcing of IC manufacturing to untrusted foundries, hardware Trojans (HT) have emerged as a major security threat and attracted much attention in FPGA trust filed. Unlike the ASIC, the FPGA trust focuses on two aspects, FPGA devices and FPGA designs. The trust of FPGA devices is the same with the trust of ASIC, and there are several surveys about it [21][22]. The trust of FPGA designs involves all steps of FPGA design flow.

FPGA trust has been investigated earlier in several articles [9][23][24][25][26]. In 1999, Hadzic et al. [23] first proposed the concept of FPGA viruses that an adversary exploits the logical or electrical attacks to create electrical conflicts causing malfunction or damaging FPGA devices. Trimberger simply discussed the trust problem in FPGAs in [9]. Chakraborty et al. [25] demonstrated that the hardware Trojan can be directly inserted by modifying the FPGA configuration bitstream. Recently, Mal-Sarkar et al. [26] presents a taxonomy of Trojan attacks in FPGA devices based on activation and payload characteristics and proposed to bypass the effect of Trojan by using the method of triple modular redundancy.

### C. Cloning

Cloning is considered to be the most common security vulnerability of volatile FPGAs. The corresponding defense techniques including watermarking, fingerprinting, encryption and PUF are proposed.

*1) Watermarking:* Digital watermarking is a candidate technology for FPGA IP protection. It embedded an encrypted watermark into the FPGA design to represent ownership. When intellectual property disputes occur, the owner will ask a trusted third party (TTP) to recover the watermark from the stolen IP core.

Lach et al. first proposed the concept of watermarking FPGA in [27], and then a number of FPGA watermarking techniques, which embed watermarks at behavioral level [28], netlist level [29], physical level [30][31] and bit-stream level [32], have been proposed for FPGA IP protection. Constraint-based watermarking [28] is a kind of typical watermarking technique for IP protection. The key idea involves making use of a number of satisfiability (SAT) issues in IP development process to transform the embedded watermark into a set of additional constraints, then limiting the solution of the SAT problem into a smaller space when they are added to the constraints, next, generating a unique design with a watermark. However, this watermarking technique has the verifiable problem that it is hard to directly verify the watermark distributed at the lower abstract levels of FPGA design flow, particularly after the IP core is synthesized into bit-stream. To address the problem, the concept of power watermarking was proposed by Ziener et al. [33][34][35]. They proposed to convert functional LUTs to LUT-based RAMs or shift registers that prevents the deletion of watermarks due to optimization [29], and detect the watermarks at the power supply pins of FPGA [33][34][35]. Usually, watermarking techniques need a trusted third party to verify the watermark when intellectual property disputes occur. However, it would be more practical if the embedded watermarks can be publicly detectable. Qu [36] proposed a publicly detectable watermarking method that embeds a separate public watermark for public verification. However, this method is not suitable to the FPGA design because it is essentially a bit-file where the embedded public watermark can be moved or covered by an adversary due to the disclosure of watermark locations in public verification process. A zero-knowledge protocol-based publicly detectable watermarking method [37] was proposed for FPGA watermarking to resolve the security issue of leaking sensitive information in the process of public verification. However, the method strongly depends on the watermark embedding methods proposed in [30][31] that require reverse engineering the bitstream to extract the watermarking in the public verification, which is obviously impractical. Hence, it is still a big challenge to

develop publicly detectable watermarking techniques without the verifiable problem.

*2) Fingerprinting:* As we discussed above, watermarking is to embed the same mark to all IPs. The distributed IP instances to different buyers are identical. Hence, the watermarking technique cannot trace the source of IP infringement. Fingerprinting is such technique that embeds a watermark along with the signature of IP buyers to present the intellectual property. Hence, the fingerprinting technique can trace the source of illegally distributing IPs. Several fingerprint methods have been proposed [38][39][40][41].

*3) Encryption:* Bitstream encryption is to encrypt the configuration bit stream by EDA tool and then decrypt it using standardized secure decryption algorithms with a nonvolatile key stored in the secure memory in an FPGA when it is loaded into an FPGA. Bit-stream encryption is one of the most popular intellectual property protection techniques against direct cloning of single large FPGA configurations for high-end FPGA devices. It has been attracted extensive attention in academia and also widely used in high-end commercial FPGAs.

In the industry domain, there are some high-end FPGA series supporting bitstream encryption. For example, Xilinx Virtex-II FPGA series [42] store the decryption key in a dedicated battery-backed SRAM or Spartan/Virtex-6 series store the decryption key in a one-time programmable eFuse register to support the bitstream encryption technique; Altera also offer bit stream encryption for their FPGAs such as the Stratix-II series [43]. In addition, some bitstream authentication methods [44][45] are also proposed.

In the academic community, Gneysu et al. [46] proposed to use the secondary secure key register and the authenticated bitstream encryption with minor modification to the current FPGA technology to protect FPGA bitstreams from cloning. A public-key-based protocol was designed between IP providers, FPGA-based system developers and a TTP to handle key exchange and installation in the symmetric-key-decryption engines [47]. With the increasing of the capacity of FPGA, IP cores have been used increasingly to control design complexity. Hence, multiple IP protection is needed. However, most of encryption-based techniques can only protect the single large FPGA configurations. Encryption-based protection method was proposed to protect multiple IPs in [48].

In addition, It is worth noting that in current pricing model, system developers cannot determine how many FPGA devices their products have been authorized running on, or IP venders cannot determine how many times IP cores have been programmed into FPGAs, which forced them to adopt an up-front licensing model where a customer obtains unlimited use of a design on any FPGAs for a single relatively large payment. However, the basic motivation for using an FPGA rather than an ASIC is to trade off a higher per-unit cost to avoid a large up front NRE payment [8]. Hence, new licensing models, pay-per-device [47] or pay-per-use [8][49][50], are needed.

*4) PUF:* Silicon PUF is a popular hardware security primitive that exploits the intrinsic variation of IC manufacturing process to generate chip-unique information for various security related applications [51]. The detailed introduction about the PUF, please refer to the recent surveys [51][52].

For encryption-based methods, every FPGA had the same key on board, which implies that if an attacker has one key he can get the secret information from all FPGAs. PUF can generate chip-unique volatile key for encryption. Hence, even if an adversary obtains the key in an FPGA by side-channel or physical attacks, he still cannot get the secret information from other FPGAs. Additionally, traditional bitstream encryption methods introduce security vulnerabilities such as physical attacks and side channel attacks due to the permanent key storage and management, and more importantly, it is well-known that such permanent key storage scheme allows attackers to attack at any time [47]. Therefore, PUF has been proposed for FPGA hardware IP (HW-IP) and software IP (SW-IP) protection.

HW-IP cores are defined as the soft-core (synthesized from HDL) hardware modules stored in the FPGA configuration bit-stream. Guajardo et al [53] proposed a public-key (PK) cryptography-based protocols for the IP protection using the SRAM PUF. However, encryption-based methods are not appropriate for resource-limited environments. In addition, there are many low-end FPGA families (e. g., the Xilinx Spartan FPGA Family) which do not support bitstream encryption. Zhang et al. [47][54] proposed a binding method that provides the lightweight IP protection for FPGAs and enables the pay-per-device pricing model.

SW-IP cores are defined as the software modules which run on a microcontroller in combination with application-specific coprocessors. The protection of the SW-IPs from cloning attacks is very similar to that of the HW-IPs. In [55], Simpson et al. introduce a scheme to protect SW-IPs against non-authorized use in reconfigurable platforms. The implementation of their scheme makes use of a PUF (they assume that an ideal PUF exists on the FPGA) and a symmetric cipher. An improved version of this scheme is subsequently proposed in [56] by Guajardo et al. Recently, Gora et al. [57] propose to protect SW-IP in FPGAs by binding it to a single trusted FPGA platform using a PUF in the FPGAs reconfigurable logic. Their proposal specifically protects the SW-IPs and assumes that the HW-IP cores are securely configured by other means of protection [50].

### *D. Reverse engineering*

Unlike ASICs, An FPGA bitstream is essentially a binary bitfile which is vulnerable to reverse engineering attacks. Reverse engineering can be misused to steal and/or pirate a FPGA design. For example, an adversary can extract the netlist information by reverse engineering the FPGA bitstream to steal the valuable intellectual property information, even illegally integrate it into his own system or directly sell it as an IP core. Techniques and tools have been developed to reverse engineer FPGA configuration [58][59]. Encryption (see Section III.C.3) and obfuscation are two defenses that have been proposed to thwart reverse engineering attacks.

Obfuscation is to hide the functionality and implementation of a design by inserting additional gates [60] or add redundant FSM states [19][61][62] into it. Without the correct key, the functionality of the design will not exhibit correctly. We note that the typical combinational logic obfuscation for ASICs proposed in [60] is not suitable for obfuscating FPGA designs because adversaries can infer the key bits according to the

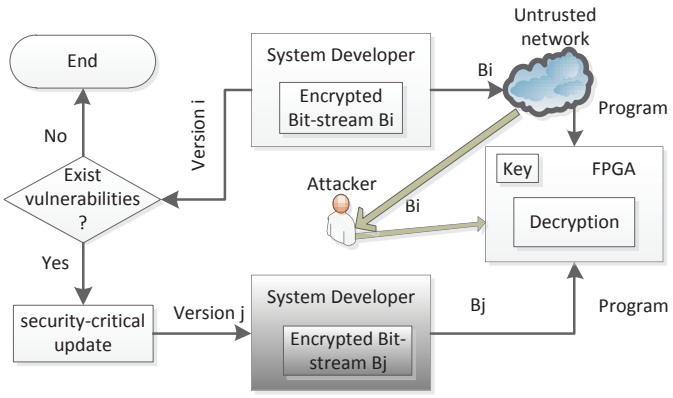


Fig. 2. Replay attacks [14].

type of inserted gates when the gate-level netlist is extracted from FPGA bitstream by reverse engineering. In order to avoid this problem, It is necessary to replace some XOR gates with XNOR gates and inverters and, similarly, replace some XNOR gates with XOR gates and inverters according to the suggestion in [60], however, which incurs high area and power overheads [63] due to the redesign of logic.

### *E. Side-channel*

Side-channel are powerful attacks that exploit the leakage of physical information when an application is being executed on a system [64]. The bitstream encryption mechanisms on Altera Stratix II [65], Xilinx Virtex-II [66] and Virtex-4/5 FPGAs [67] have been reportedly broken by side-channel attacks which statistically analyze the power consumption or electromagnetic emanation of the devices. The corresponding strategies against passive side-channel attacks were proposed [68].

### *F. Replay*

The replay attacks are particularly dangerous for FPGA-based system security because attackers can effectively preclude security-critical updates by replaying the previous FPGA configurations. Current FPGA IP protection techniques do not consider the replay attack. As shown in Fig. 2, the system developers usually need to update their FPGA-based products Bi to a new version Bj for the sake of upgrading such as fixing the vulnerabilities to protect them against security threat, which gives the attackers the chance to downgrade the system into its previous old version Bi so that they can exploit the outdated vulnerabilities to steal secret information [14]. The replay attack was first introduced by Drimer [5]. the corresponding remote update protocols [69][70][71] for bitstream encryption techniques were proposed to resist replay attacks. The solution to address replay attacks for dynamic partial reconfiguration systems was presented in [72]. Recently, reconfiguring the binding scheme proposed in [47] is also considered as a new and lightweight solution to resist FPGA replay attacks [14].

## IV. Conclusion

Nowadays, FPGAs have been widely used in the computing acceleration, communication and other areas due to the continuous improvement in quality and the decrease of production cost. The security and trust of FPGA-based system have attracted much attention. This paper provides a comprehensive survey of the security and trust issues related to FPGA-based systems from the market perspective. For each issue, the state-of-the-art defenses are also elaborated.

## Acknowledgment

This work is supported by the scholarship from China Scholarship Council (CSC) under Grant No.201306130042, the National Natural Science Foundation of China under Grant No. 61228204, and a MURI award from AFOSR.

## References

[1] W. Wolf, FPGA-Based System Design. Prentice Hall, 2004.

[2] I. Kuon, R. Tessier, and J. Rose, "FPGA Architecture: Survey and Challenges," *Found. Trends Electron. Des. Autom.*, vol. 2, no. 2, pp. 135-253, 2007.

[3] D. Chen, J. Cong, and P. Pan, "FPGA Design Automation: A Survey," *Found. Trends Electron. Des. Autom.*, vol. 1, no. 3, pp. 195-334, 2006.

[4] T. Wollinger, J. Guajardo, and C. Paar, "Security on FPGAs: State-of-the-art implementations and attacks," *ACM Trans. Embed. Comput. Syst.*, vol. 3, no. 3, pp. 534-574, Aug. 2004.

[5] S. Drimer, "Volatile FPGA design security - a survey," University of Cambridge, 2008.

[6] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "FPGA-oriented Security," Springer, pp. 1-38.

[7] B. Badrignans, F. Devic, L. Torres, G. Sassatelli, and P. Benoit, *Security Trends for FPGAs*. Dordrecht: Springer Netherlands, 2011.

[8] T. Kean, "Cryptographic rights management of FPGA intellectual property cores," in *FPGA*, 2002, pp. 113-118.

[9] S. Trimberger, "Trusted Design in FPGAs," in *DAC*, 2007, pp. 5-8.

[10] J. Gu, G. Qu, and Q. Zhou, "Information hiding for trusted system design," in *DAC*, 2009, pp. 698-701.

[11] T. Huffmire, B. Brotherton, T. Sherwood, R. Kastner, T. Levin, T. D. Nguyen, and C. Irvine, "Managing Security in FPGA-Based Embedded Systems," *IEEE Des. Test Comput.*, vol. 25, no. 6, pp. 590-598, Nov. 2008.

[12] S. Trimberger and J. Moore, "FPGA Security," in *DAC*, 2014, pp. 1-4.

[13] F. Durvaux, S. Kerckhof, F. Regazzoni, and F. Standaert, "A Survey of Recent Results in FPGA Security and Intellectual Property Protection," in *Secure Smart Embedded Devices, Platforms and Applications*, K. Markantonakis and K. Mayes, Eds. New York, NY: Springer New York, 2014, pp. 201-224.

[14] J. Zhang, Y. Lin, and G. Qu, "Reconfigurable Binding against FPGA Replay Attacks," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 2015.

[15] F. Koushanfar, G. Qu, and M. Potkonjak, "Intellectual Property Metering", *Information Hiding Workshop*, pp. 81-95, 2001.

[16] Y. Alkabani, F. Koushanfar, N. Kiyavash, and M. Potkonjak, "Trusted integrated circuits: A nondestructive hidden characteristics extraction approach," in *Information Hiding*, 2008, pp. 102-117.

[17] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *HOST*, 2008, pp. 67-70.

[18] F. Koushanfar, "Provably Secure Active IC Metering Techniques for Piracy Avoidance and Digital Rights Management," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 1, pp. 51-63, 2012.

[19] Y. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in *USENIX Security*, 2007, pp. 291-306.

[20] F. Koushanfar, "Integrated circuits metering for piracy protection and digital rights management," in *GLSVLSI*, 2011, pp. 449-454.

[21] R. S. Chakraborty, S. Narasimhan, and S. Bhunia, "Hardware Trojan: Threats and emerging solutions,' in *IEEE International High Level Design Validation and Test Workshop*, 2009, pp. 166-171.

[22] M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 10-25, Jan. 2010.

[23] I. Hadzic, S. Udani, and J. M. Smith, "FPGA Viruses," in *FPL*, 1999, pp. 291-300.

[24] M. Patterson, A. Mills, R. Scheel, J. Tillman, E. Dye, and J. Zambreno, "A multi-faceted approach to FPGA-based Trojan circuit detection," in *VTS*, 2013, pp. 1-4.

[25] R. S. Chakraborty, I. Saha, A. Palchaudhuri, and G. K. Naik, "Hardware Trojan Insertion by Direct Modification of FPGA Configuration Bitstream," *IEEE Des. Test*, vol. 30, no. 2, pp. 45-54, Apr. 2013.

[26] S. Mal-Sarkar, A. Krishna, A. Ghosh, and S. Bhunia, "Hardware trojan attacks in FPGA devices," in *GLSVLSI*, 2014, pp. 287-292.

[27] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, "Signature hiding techniques for FPGA intellectual property protection," in *ICCAD*, pp. 186-189, 1998.

[28] A. B. Kahng, et al., "Constraint-Based Watermarking Techniques for Design IP Protection," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 20, no. 10, pp. 1236-1252, 2001.

[29] M. Schmid, D. Ziener, and J. Teich, "Netlist-level IP protection by watermarking for LUT-based FPGAs," in *FPT*, 2008, pp. 209-216.

[30] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, "Robust FPGA intellectual property protection through multiple small watermarks," in *DAC*, 1999, pp. 831-836.

[31] A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, "Watermarking techniques for intellectual property protection," in *DAC*, 1998, pp. 776-781.

[32] J. Zhang, Y. Lin, Q. Wu, and W. Che, "Watermarking FPGA Bitfile for Intellectual Property Protection," *Radioengineering*, pp. 764-771, 2012.

[33] D. Ziener and J. Teich, "Power Signature Watermarking of IP Cores for FPGAs," *J. Signal Process. Syst.*, vol. 51, no. 1, pp. 123-136, 2007.

[34] D. Ziener and J. Teich, "FPGA core watermarking based on power signature analysis," in *FPT*, pp. 205-212, Dec. 2006.

[35] D. Ziener, F. Baueregger, and J. Teich, "Multiplexing methods for power watermarking," in *HOST*, pp. 36-41, Jun. 2010.

[36] G. Qu, "Publicly detectable watermarking for intellectual property authentication in VLSI design," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 21, no. 11, pp. 1363-1368, Nov. 2002.

[37] D. Saha and S. Sur-Kolay, "Secure Public Verification of IP Marks in FPGA Design Through a Zero-Knowledge Protocol," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 20, no. 10, pp. 1749-1757, Oct. 2012.

[38] J. Lach, W. Mangione-Smith, and M. Potkonjak, "Fingerprinting digital circuits on programmable hardware," in *Information Hiding*, 1998, pp. 16-31.

[39] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, "Fingerprinting techniques for field-programmable gate array intellectual property protection," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 20, no. 10, pp. 1253-1261, 2001.

[40] G. Qu and M. Potkonjak, "Fingerprinting intellectual property using constraint-addition," in *DAC*, 2000, pp. 587-592.

[41] C. Chang and L. Zhang, "A Blind Dynamic Fingerprinting Technique for Sequential Circuit Intellectual Property Protection," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 33, no. 1, pp. 76-89, 2014.

[42] "Using high security features in Virtex-II series FPGAs (v1.0)," Xilinx App. Note 766, 2004.

[43] "Design security in Stratix III devices (v1.5)," Altera White Paper 01010, 2009.

[44] S. Trimberger, J. Moore, and W. Lu, "Authenticated encryption for FPGA bitstreams," in *FPGA*, 2011, pp. 83-86.

[45] Y. Hori and A. Satoh, "Bitstream encryption and authentication with AES-GCM in dynamically reconfigurable systems," in *FPL*, 2008, pp. 23-28.

[46] T. Guneysu, B. Moller, and C. Paar, "Dynamic Intellectual Property Protection for Reconfigurable Devices," in *FPT*, 2007, pp. 169-176.

[47] J. Zhang, Y. Lin, Y. Lyu, and G. Qu, "A PUF-FSM Binding Scheme for FPGA IP Protection and Pay-per-Device Licensing," *IEEE Trans. Inf. Forensics Secur.*, 2015.

[48] S. Drimer and T. Gneysu, "Protecting multiple cores in a single FPGA design," http//www.cl.cam.ac.uk/ sd410/papers/protect many cores.pdf, pp. 1-21, 2008.

[49] K. Kepa, F. Morgan, and K. Kosciuszkiewicz, "IP protection in Partially Reconfigurable FPGAs," in *FPL*, 2009, pp. 403-409.

[50] R. Maes, D. Schellekens, and I. Verbauwhede, "A Pay-per-Use Licensing Scheme for Hardware IP Cores in Recent SRAM-Based FPGAs," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 1, pp. 98-108, Feb. 2012.

[51] J. Zhang, G. Qu, Y. Lv, and Q. Zhou, "A Survey on Silicon PUFs and Recent Advances in Ring Oscillator PUFs," *Journal of Computer Science and Technology*, vol. 29, no. 4, pp. 664-678, Jul. 2014.

[52] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," *Proc. IEEE*, pp. 1-16, 2014.

[53] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "Physical Unclonable Functions and Public-Key Crypto for FPGA IP Protection," in *FPL*, 2007, pp. 189-195.

[54] J. Zhang, Y. Lin, Y. Lyu, G. Qu, R. C. C. Cheung, W. Che, Q. Zhou, and J. Bian, "FPGA IP protection by binding Finite State Machine to Physical Unclonable Function," in *FPL*, 2013, pp. 1-4.

[55] E. Simpson and P. Schaumont, "Offline Hardware / Software Authentication for Reconfigurable Platforms," pp. 311-323, 2006.

[56] J. Guajardo, S. S. Kumar, G. Schrijen, and P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection," in *CHES*, 2007, pp. 63-80.

[57] M. a. Gora, a. Maiti, and P. Schaumont, "A Flexible Design Flow for Software IP Binding in FPGA," *IEEE Trans. Ind. Informatics*, vol. 6, no. 4, pp. 211-218, 2010.

[58] J.-B. Note and E. Rannaud, "From the bitstream to the netlist," in *FPGA*, 2008, p. 264.

[59] F. Benz, A. Seffrin, and S. A. Huss, "Bil: A tool-chain for bitstream reverse-engineering," in *FPL*, 2012, pp. 735-738.

[60] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending Piracy of Integrated Circuits," in *DATE*, 2008, pp. 1069-1074.

[61] Y. Alkabani, F. Koushanfar, and M. Potkonjak, "Remote activation of ICs for piracy prevention and digital right management," in *ICCAD*, 2007, pp. 674-677.

[62] R. S. Chakraborty and S. Bhunia, "RTL Hardware IP Protection Using Key-Based Control and Data Flow Obfuscation," in *Int. Conf. VLSI Des.*, pp. 405-410, Jan. 2010.

[63] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *DAC*, 2012, pp. 83-89.

[64] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *CRYPTO*, 1999, pp. 388-397.

[65] A. Moradi, D. Oswald, C. Paar, and P. Swierczynski, "Side-channel attacks on the bitstream encryption mechanism of Altera Stratix II," in *FPGA*, 2013, p. 91.

[66] A. Moradi, A. Barenghi, T. Kasper, and C. Paar, "On the vulnerability of FPGA bitstream encryption against power analysis attacks," in *CCS*, 2011, p. 111.

[67] A. Moradi, M. Kasper, and C. Paar, "On the Portability of Side-Channel Attacks - An Analysis of the Xilinx Virtex 4, Virtex 5, and Spartan 6 Bitstream Encryption Mechanism," *Int. Assoc. Cryptologic Res.*, 2011.

[68] A. Bogdanov, A. Moradi, and T. Yalcin, "Efficient and side-channel resistant authenticated encryption of FPGA bitstreams," in *International Conference on Reconfigurable Computing and FPGAs*, 2012, pp. 1-6.

[69] S. Drimer and M. G. Kuhn, "A Protocol for Secure Remote Updates of FPGA Configurations," in *Arc*, 2009, pp. 50-61.

[70] B. Badrignans, R. Elbaz, and L. Torres, "Secure FPGA configuration architecture preventing system downgrade, in *FPL*, 2008, pp. 317-322.

[71] F. Devic, L. Torres, and B. Badrignans, "Secure Protocol Implementation for Remote Bitstream Update Preventing Replay Attacks on FPGA," in *FPL*, 2010, pp. 179-182.

[72] F. Devic, L. Torres, J. Crenne, B. Badrignans, and P. Benoit, "SecURe DPR: Secure update preventing replay attacks for dynamic partial reconfiguration," in *FPL*, 2012, pp. 57-62.