In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy.

A proposal for the secure activation and licensing of FPGA IP cores

Domenico Amelino[2], Mario Barbareschi[12], and Alessandro Cilardo[12]

[1] DIETI - Department of Electrical Engineering and Information Technologies
University of Naples Federico II.
[2] CeRICT scrl - Centro Regionale Information Communication Technology

Abstract

The fabless business model is leading to intellectual property (IP) based design for System-on-chip devices, involving both the field programmable gate array (FPGA) and application specific integrated circuit (ASIC) technology. The main advantage is essentially the decoupling of the system integration phase from the development of the single cores. The use of third-party IPs, however, raises many challenges, especially related to the security of the manufactured devices, as the dynamic installation/activation of new functions makes it more difficult to track the distribution and use of licensed IPs. In that respect, an effective solution involves the online interaction between each end user's device and the IP provider, but this online tracking comes at the price of compromised user's privacy. Bearing in mind this consideration, the work proposes the adoption of a concept borrowed from the trusted computing area, the so-called Direct Anonymous Attestation, to enable remote IP licensing and activation mechanisms while fully preserving the anonymity of the end user, i.e., making it impossible to infer the user's identity from its behaviour as seen by the activation server. The main contribution of this paper is the definition of an ad-hoc protocol, called Remote Anonymous Activation Protocol (RAAP), as well as a proof-of-concept implementation on a commercial target device, which encompasses an FPGA and a general purpose processing system.

1 Introduction

In the recent years, the emerging role of FPGA devices and their in-field reconfigurability have introduced new security challenges, from both the Intellectual Property (IP) provider's and the user's point of view. In particular, thanks to hardware reconfigurability which allows devices to be dynamically extended with new hardware functions (the IPs), the device manufacturers and the IP providers become *two separate roles*, creating new needs for the IP providers to track the distribution and use of their IPs deployed on third-party devices.

To defend the emerging business models enabled by such new scenarios, the technical literature introduced a number of solutions for guaranteeing IP security in hardware-reconfigurable systems. For example, Simpson et al. propose in [22] a scheme for the offline autentication of IP FPGA cores based on silicon Physically Unclonable Functions (PUFs) [18]. Similarly, [27] presents an activation mechanism based on the binding of a partial bitstream to an FPGA device. Parrilla et al. [19] describes an activation mechanism using Elliptic Curve Cryptography (ECC) and PUFs. Interestingly, many proposals rely on a few concepts borrowed from the so-called *trusted computing* area, e.g. the Trusted Platform Module (TPM) specified by the Trusted Computing Group (TCG) consortium [1] along with the related Digital Right Management (DRM) mechanisms. The work in [20] presents Algodone Smart Lock and an activation mechanism relying on PUFs as a device signature. Further examples include: Serecon [14] which targets IP protection in dynamically hardware-reconfigurable trusted platforms and identifies the essential roles in the infrastructure, i.e. the system integrator, the trusted authority, the

FPGA fabric vendor, etc; Couture et al. propose in [11] an extension of licensing mechanisms to FPGA components augmenting the system with a secure non-volatile memory (NVM) and a tamper-resistant unique identifier; the papers [13, 12] illustrate a volume licensing scheme for FPGA bitstreams, extended to the case of multiple cores within one FPGA; Maes et al. in [17] introduce a pay-per-use licensing scheme protecting individual FPGA IP cores; Cilardo et al. in [10] identify the roles involved in the secure IP distribution process for FPGAs and introduce a cryptographic protocol ensuring the confidentiality and the trustworthiness of partial bitstreams dynamically downloaded to the user's device.

Conversely, in this work, we are interested in new forms of interactions enabled by Trusted Computing, not limited to the use cases covered by the above works. In particular, we focus on *remote attestation*, a process specified by the Trusted Computing architecture to establish trusted relationships between devices and third parties, letting the device prove to a remote verifier that the platform has a valid configuration. While the initial TCG approach relying on a *Privacy Certification Authority* (CA) [2] had inherent limitations, because the Privacy CA turned out to be both a performance and a security bottleneck, the so-called *Direct Anonymous Attestation* (DAA) [4] establishes a trusted relation among parties without the online participation of a trusted party, while preserving the user's privacy.

Based on group signature [7] (which essentially enables users to prove that they are part of a trusted group), the first proposal for DAA [1] relied on RSA. For efficiency reasons, several solutions later tweaked the basic protocol to reduce the computational load of DAA operations [15, 16]. In particular, [5] introduced a DAA scheme based on ECC and bilinear maps, labelled ECC-DAA. Based on this idea, a few contributions tried to limit the computational load incurred by the fundamental device-side component, e.g. the Trusted Platform Module (TPM), which embraces all the security-critical operations [8, 6]. In fact, the TPM 2.0 specification draft [3] includes multiple ECC-DAA schemes. Several works particularly addressed the implementation of DAA on security-enabled embedded platforms, such as ARM TrustZone, providing a secure perimeter within the platform. Reference [23] presents a lightweight anonymous authentication scheme for embedded devices. Similarly, [28] proposes a DAA framework for mobile platforms. The work in [25] describes four ECC-based DAA implementations. Reference [26] also introduces a DAA scheme based on TrustZone. A few works [24, 9] present a comparison of different DAA solutions, which may drive the choice of DAA schemes for a given infrastructure/platform.

Although this work does not deal with the DAA algorithms by themselves, it is useful to summarize the essential concepts behind the scheme. The DAA requires three players, the Issuer, granting authentication credentials, the Device, made of a Host and a Trusted Module (TM), which are enabled to anonymously attest their own platform, and the Verifier, checking whether the signature generated by a Device belongs to the DAA group and, consequently, it can be considered trustworthy. The whole process preserves the user privacy since no player can associate a signature with a user identity, except the Issuer. The scheme also covers the case that a compromised (i.e. a *rogue*) TPM attempts to sign a message, by employing a revoking mechanism. In essence, the DAA protocol is structured in four main phases: Setup, Join, Sign, and Verify. In the Setup phase, the Issuer generates a DAA public key and the Issuer secret key. In the Join phase, the Device uses its TM to generate a secret value, then securely stored, and a public value sent to the Issuer, which in turn generates a credential for the Device. In the Sign phase, the Device generates a signature to attest its platform trustworthiness to the Verifier. Each sign operation is masked with a *nonce* value to avoid replay attacks. In the Verify phase, the Verifier checks the received signature and, after looking up a revocation list provided by the infrastructure, it establishes whether the device is trusted or not.

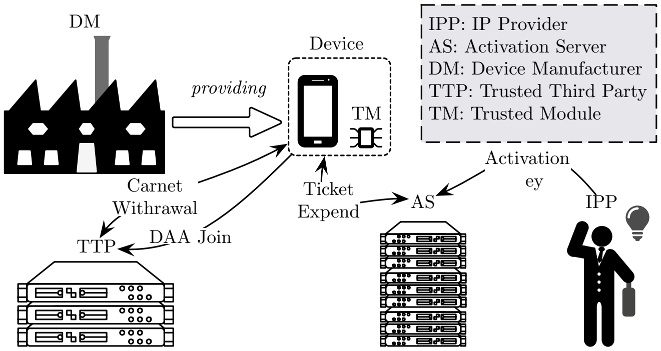

Figure 1: Interactions among players of the proposed RAAP scheme.

The work described in this paper borrows a few concepts from the DAA scheme and applies them to remote IP licensing and activation. It assumes that the end-user devices provide an environment for run-time download and activation of FPGA-based IP cores, a process mediated by an *Activation Server* (AS) in charge of managing the IP cores and related licensing. The paper relies on an ad-hoc protocol, called *Remote Anonymous Activation Protocol* (RAAP), regulating the IP distribution and activation processes. Based on DAA, RAAP preserves the full anonymity of the end user, i.e. it makes it impossible for an Activation Server to track the user's behaviour over the product lifecycle. To confirm the computational feasibility of RAAP, the paper also presents a demo implementation relying on a medium-size FPGA-based embedded platform.

2 Remote Anonymous Attestation Protocol

In this section we define the underlying infrastructures and main roles involved in the activation protocol. First, we assume to have a third-trusted party (TTP), which allows trustworthy communications and attestations among other involved parties. It is the only party getting access to the full user's identity. The other players are totally oblivious to the user's identity, such as the Device Manufacturer (DM), who integrates the IPs and a Trusted Module (TM) together in the manufactured device, the IP core Providers (IPPs), which guarantee full compatibility with the issued IP core design specification and other non-functional requirements, and the Activation Server, who is responsible for the licensing mechanism.

From the user-side, the device is made of the host, i.e., the hardware/software environment within the user's device, and the TM, which deals with IP core activation tasks. Figure 1 shows an overview of the players and the interactions among them, described below.

The main objective of the RAAP protocol is to realize secure remote activation of an IP Core keeping the user's identity hidden from all players but the TTP. The mechanism allows also the monitoring of the number of activations of a specific IP Core. Considering the DAA, presented in Figure 1, the TTP coincides with the DAA issuer, in that it provides group sign credentials to the end user's device. Once the TM of a device gets the DAA credentials, it is able to contact the AS in an anonymous way. The TTP issues *activation tickets* used to unblock an IP for a finite number of activations. The end user can acquire one or more such tickets (a *carnet* of tickets) from TTP and spend them at the ASs. Acquisition may be accomplished on a subscription basis (pay-per-use scheme), or it may correspond to the registration of the user

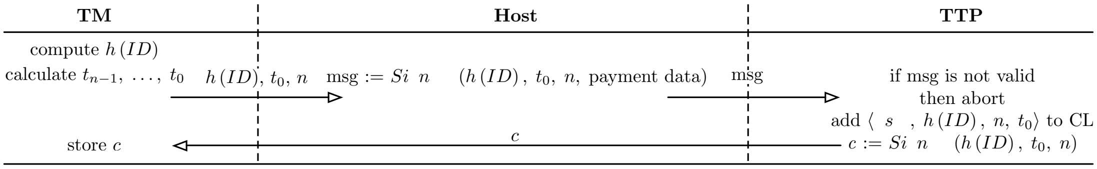
TM
Host
TTP
compute $h\,(ID)$
calculate $t_{n-1},\ \ldots,\ t_0$
$h\,(ID),\ t_0,\ n$
msg := $Si\ \ n\ \ \ (h\,(ID)\,,\ t_0,\ n,$ payment data)
msg
if msg is not valid
then abort
add $\langle\ \ s\ \ \ ,\ h\,(ID)\,,\ n,\ t_0\rangle$ to CL
$c := Si\ \ n\ \ \ \ (h\,(ID)\,,\ t_0,\ n)$
c
store c

Figure 2: Ticket Withdrawal Phase.

platform to the infrastructure for metering purposes. It must be guaranteed that each ticket can be spent once and anonymously.

The AS, that acts as the Verifier of the DAA, owns the activation key k for the IP cores, used to derive licenses requested by hosts. The AS verifies the tickets authenticity before issuing a license to the host.

On the user device side, the activation process has to be managed by the TM, which stores the tickets since the host is executed on a non-trusted environment. In the proposed scheme, the tickets coincide with the links of a *hash chain*. When the user needs to activate a specific IP Core, the TM generates a chain of elements, sending to the TTP only the last element of the chain, while the remaining elements are either securely stored or regenerated on-demand. Depending on the correctness of the information, the TTP releases the required tickets for the activations. Once the User acquires the tickets, it can spend them to the AS, which generates and sends the license for the user.

Below we describe the details of the RAAP phases, namely the ticket withdrawal and ticket expend.

2.1 Ticket Withdrawal

The Withdrawal phase, shown in Figure 2, involves the Device and the TTP, where the TM inside the Device generates a hash chain whose length, specified by the Host, is basically established as the number of requested activations. Let n, t_{n-1}, and t_0 be, respectively, the number of activations which can be acquired, the first element of the chain, and the last element, such that $h^{n-1}\,(t_{n-1}) = t_0$. The hash function involved in the generation of the chains is provided by the TM, while t_{n-1} is a random-salted value derived from the IP core ID.

Once the hash chain has been completed, the Host gets t_0 and a blinded value of the IP core identifier, i.e. $h\,(ID)$ from the TM and, through an authenticated communication channel, forwards such information to the TTP, the amount of required activations n, and the associated payment data, in case the user has to give a proof of a payment transaction. The TTP verifies the message and payment data with the user identity. In case of success, the TTP adds a new entry in the carnet list (CL), a list that contains the association among the user's identity, $h\,(ID)$, t_0, and the number of tickets n. The TTP is able to retrieve the ID since either the IPP or the DM communicate the IDs of every manufactured device. Subsequently, the TTP creates the carnet of tickets c, by signing the $h(ID), t_0$, and n values with its private key bsk. Then the TTP forwards the carnet to the Host, which has to be stored within the TM's secure perimeter.

2.2 Ticket Expend

This phase is depicted in Figure 3 the purchased carnet of tickets can be spent by the Host to activate a specific IP instance n times. Let i be the index of the first available ticket in

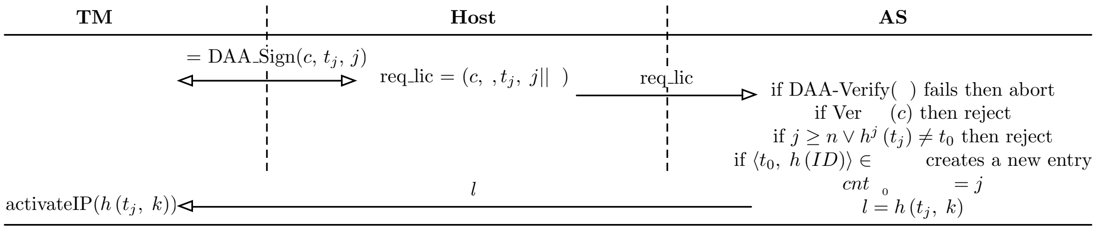
TM
Host
AS
= DAA_Sign(c, t_j, j)
req_lic = (c, , t_j, j||)
req_lic
if DAA-Verify( ) fails then abort
if Ver (c) then reject
if j ≥ n ∨ h^j (t_j) ≠ t_0 then reject
if ⟨t_0, h (ID)⟩ ∈ creates a new entry
cnt_0 = j
l = h (t_j, k)
l
activateIP(h (t_j, k))

Figure 3: Ticket Expend Phase.

the carnet, i.e. t_i, with $i \neq 0$. Then, let m be the number of tickets that the user wants to spend in a single interaction with the AS. By means of the ticket expend phase, the user requests m licenses to the AS. Hence, the host forwards to the AS a message containing the carnet c, t_j, and j, such that $j = i + m - 1$, where j indicates the counter of the tickets which have been actually spent at the AS. Indeed, the index of the next available ticket in the carnet turns out to be $j + 1$. Exploiting the DAA sign operation, the AS is able to authenticate the message in an anonymous way. If the verification succeeds, the AS retrieves t_0, n, and $h(ID)$ by authenticating the carnet c using the Issuer (i.e. TTP) public key, *bpk*.

Then, the AS has to check if j is greater than the maximum number of purchased activations n and the correctness of the value t_j. The AS also checks if the carnet has already been used in previous requests by looking up a local carnet list (LCL) and then updating the counter of the issued licenses. If every check succeeds, it calculates $l = h\left(t_j, k\right)$, where k is the activation code for the specific type of IP core involved, shared between AS and IPP. This value is representative of m licenses useful for carrying out m IP core activations at the TM.

3 Case Studies

In this section, we provide a prototype implementation of the RAAP scheme on a hybrid target user device featuring an FPGA-accelerated SoC. Indeed, the choice of the hardware components has a central role for the support of performance and security critical aspects. Our main aim in this paper is to demonstrate the feasibility of the RAAP scheme particularly addressing the portion of the protocol running on the user device. We firstly describe the hardware and software test setting, based on a customizable and efficient hardware library ported to the target platform. Then, we present the overheads incurred by our scheme detailing the time- and memory-critical part of the process. We also propose the integration of custom hardware accelerators to improve the system performance on low/medium-complexity target devices, where pure software implementations might pose a limitation.

3.1 Implementation Details

As detailed in the previous sections, we consider three main entities: the TTP, the AS, and the End User. In our experiments, we simulate TTP and AS by means of a desktop PC: in a real scenario they are remote high-end servers. For the end-user device, we use a Zedboard Zynq-7000 Development Board. This device features an ARM-A9 MPCore processing system (PS) together with a reconfigurable part based on an FPGA fabric, possibly hosting IP hardware cores managed through a license-based activation mechanism. For the experimental tests, we connected the ZedBoard via a USB cable to the PC, which in turn plays the role of the TTP and the AS.

We adopt the MIRACL software library [21] for implementing the RAAP scheme on the bare metal ARM processor and on the PC for simulating the TTP and the AS. This library provides optimized implementations of cryptographic primitives, particularly Elliptic Curve Cryptography operations and supports a wide range of platforms such as x86-x64 Intel and ARM. The following evaluations do not consider a separation between host and TM: our experiments mainly aim at demonstrating the computational feasibility of the remote attestation and licensing mechanism of the RAAP scheme.

3.2 Experimental Results: Software Implementation

As explained in the previous sections, RAAP is composed of two main steps: Ticket Withdrawal and Ticket Expend. We evaluate the time overhead of the execution on ZedBoard development board and each average experimental result is taken over 20 test-runs.

Based on the interactions required by RAAP, we relied on the following combined use of different cryptographic building blocks. Each player of the protocol uses 1024-bit RSA cryptography for authentication and message exchange. Furthermore, since RAAP partially relies on an anonymous group signature scheme, as required by the DAA anonymous attestation technique, we need to support ECC. In particular, we adopt a 128-bit BN curve DAA ECC scheme for allowing users to acquire a trustworthy group signature from the TTP. Last, the hash chain is generated by means of a SHA-256 cryptographic hash function.

As discussed in Section 1 the DAA-Join phase allows a device to acquire group credentials from a trusted party used to attest End User Device trustworthiness to other parties. It is important to notice that this operation is sporadically executed and it is not of interest for our scopes.

The following results regard only the software-based implementation. Further details about the introduction of hardware acceleration are given in Section 3.3.

As described before, the Ticket Withdrawal executes a blind signature, a hash chain generation, a request signature, and a carnet verification. This phase is executed each time the user needs to acquire licenses for activating IP Cores. For hash chain generation, we consider a hash chain size of 1000 elements. The blind signature, complying with NIST specification, is characterized by an execution time of 29.7 ms. The overall Ticket Withdrawal average execution time is 2653 ms. As regards the generation of the hash chain, the overhead becomes significantly high only when the chain size reaches 10,000 elements: below this value, the SHA-256 introduces a limited time overhead, around 500 ms. The above time overheads can be considered negligible since the corresponding operations occur infrequently compared to the application lifecycle.

The Ticket Expend phase is dominated by the DAA-Sign. The DAA-Sign implemented on the ZedBoard platform is characterized by an average elapsed time of about 833 ms for each license acquisition offering a reasonable security level (128-bit) and a reasonable time overhead for an embedded device. It is worth noting that the presented results do not take into account the time overhead caused by network communications as well as the time required for I/O operations.

3.3 Custom hardware accelerator

As highlighted by the results above, the cryptographic operations behind the RAAP scheme involve a significant computational load due to large operands and complex mathematical operations.

Table 1: Execution time of some primitives in MIRACL in DAA Join and Sign phases

	DAA-Join (μs)	DAA-Sign (μs)	Cumulative Execution time (μs)
mr_padd	7348 (5%)	6862,6 (2,4%)	1381,3
mr_psub	5358,3 (4%)	5961,4 (2,1%)	11291
muldiv	9138 (7%) (7%)	50454 (18%)	59018
red_c	68171(49,5%)	107458,5(36,5%)	193194

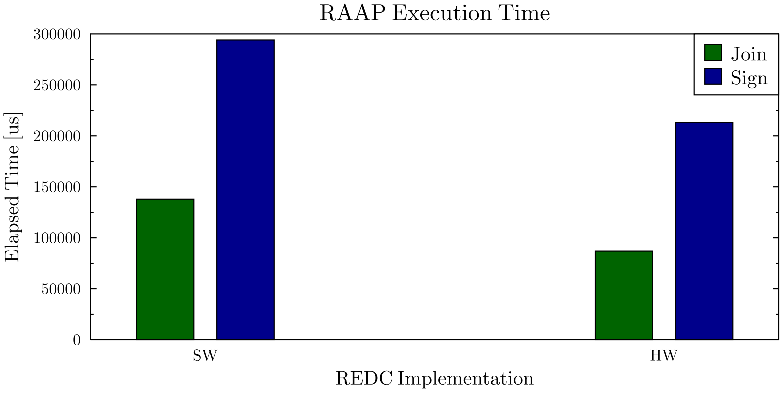

Figure 4: Comparison between elapsed time in DAA steps adopting software and hardware *red_c* implementations

For this reason, we profiled the most performance-critical operations, shown in Table 1. The table illustrates the impact of the main MIRACL primitives on performance, specifying the cumulative time spent by each function during the DAA-Join and DAA-Sign phases. We identify the Montgomery Reduction, called *red_c* function in the code, as the main performance hotspot of the implementation. In particular, the *red_c* contributes to DAA-Join and DAA-Sign execution time respectively by 49.4% and 36.5%. The operation computes the product $Y = X \cdot R^{-1} \bmod N$ and can be profitably migrated to hardware.

In order to improve the system performance, we evaluate the integration of hardware acceleration for the Montogomery Reduction in place of the software *red_c*. We designed a hardware component as an AXI4 slave peripheral which wraps a dedicated data path for the *red_c*. The datapath handles input binary strings up to 576 bits. The input and output protocols, as well as the synchronization mechanism, are arbitrated by a software driver available as a bare metal function, seamlessly integrated within the MIRACL library in place of the software *red_c*. The implementation of the hardware *red_c* yields a resource occupation of 714 Look-up Tables, 16 $DSP-48$ blocks, 893 Flip-Flops, 4 Block-RAMs, and 560 Data Memory RAMs. Arithmetic operators (multiplication and addition) are pipelined and customized for the accelerated operation, leading to a clock speed of 142.33 MHz. Figure 4 shows a comparison between DAA-Join and DAA-Sign execution times adopting software and hardware implementation of the *red_c* function. Hardware acceleration for Montgomery reduction gains a considerable speed-up of 37% and 27% respectively for DAA-Join and DAA-Sign. In particular, the DAA-Sign is the core operation of RAAP Ticket Expend, showing a considerable improvement in the target system performance. The measurements with the hardware core also consider the elapsed time required to exchange data between the processing system and the core itself.

4 Conclusions

In this paper, we have shown an anonymous activation protocol based on the concept of Direct Anonymous Attestation. The solution introduces a licensing/activation protection mechanism for IP core-based Systems-on-Chip. Our proposal also allows implementing hardware metering to monitor core activations and uses. We have provided a thorough description of the protocol and a prototype implementation highlighting its suitability for lightweight user devices. The results confirm the feasibility of the proposed protocol, even considering user's devices with limited compute resources, like the one adopted for our experimental evaluation.

References

[1] TPM main part 1 design principles specification version 1.2, 2011.

[2] Trusted computing platform alliance (TCPA) main specification, version 1.1a republished as trusted computing group (TCG) main specifcation, 2011.

[3] Trusted platform module library part 1: Architecture 2.0, 2014.

[4] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145. ACM, 2004.

[5] Ernie Brickell, Liqun Chen, and Jiangtao Li. A new direct anonymous attestation scheme from bilinear maps. In *Trusted Computing-Challenges and Applications*, pages 166–178. Springer, 2008.

[6] Ernie Brickell and Jiangtao Li. A pairing-based DAA scheme further reducing TPM resources. In *Trust and Trustworthy Computing*, pages 181–195. Springer, 2010.

[7] David Chaum and Eugène Van Heyst. Group signatures. In *Advances in CryptologyEUROCRYPT91*, pages 257–265. Springer, 1991.

[8] Liqun Chen. A DAA scheme requiring less TPM resources. In *Information Security and Cryptology*, pages 350–365. Springer, 2009.

[9] Liqun Chen, Dan Page, and Nigel P Smart. On the design and implementation of an efficient DAA scheme. In *Smart Card Research and Advanced Application*, pages 223–237. Springer, 2010.

[10] Alessandro Cilardo, Mario Barbareschi, and Antonino Mazzeo. Secure distribution infrastructure for hardware digital contents. *Computers & Digital Techniques, IET*, 8(6):300–310, 2014.

[11] Nathaniel Couture and Kenneth B Kent. Periodic licensing of FPGA based intellectual property. In *Field Programmable Technology, 2006. FPT 2006. IEEE International Conference on*, pages 357–360. IEEE, 2006.

[12] Saar Drimer, Tim Güneysu, Markus G Kuhn, and Christof Paar. Protecting multiple cores in a single FPGA design. *Draft available at http://www.cl.cam.ac.uk/sd410/, written May*, 2008.

[13] Tim Guneysu, Bodo Möller, and Christof Paar. Dynamic intellectual property protection for reconfigurable devices. In *Field-Programmable Technology, 2007. ICFPT 2007. International Conference on*, pages 169–176. IEEE, 2007.

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

[15] Jiqiang Liu, Jia Zhao, and Zhen Han. A remote anonymous attestation protocol in trusted computing. In *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pages 1–6. IEEE, 2008.

[16] Li Lixin, Li Chaoling, and Zhou Yanzhou. A remote anonymous attestation scheme with improved privacy CA. In *2009 International Conference on Multimedia Information Networking and Security*, volume 1, pages 153–157. IEEE, 2009.

[17] Roel Maes, Dries Schellekens, and Ingrid Verbauwhede. A pay-per-use licensing scheme for hardware IP cores in recent SRAM-based FPGAs. *Information Forensics and Security, IEEE Transactions on*, 7(1):98–108, 2012.

[18] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

[19] Luis Parrilla, Encarnación Castillo, Diego P Morales, and Antonio García. Hardware activation by means of PUFs and elliptic curve cryptography in field-programmable devices. *Electronics*, 5(1):5, 2016.

[20] Jerome Rampon, Renaud Perillat, Lionel Torres, Pascal Benoit, Giorgio Di Natale, and Mario Barbareschi. Digital right management for IP protection. In *VLSI (ISVLSI), 2015 IEEE Computer Society Annual Symposium on*, pages 200–203. IEEE, 2015.

[21] Micheal Scott. Multiprecision integer and rational arithmetic cryptographic library (MIRACL), 2015.

[22] Eric Simpson and Patrick Schaumont. Offline hardware/software authentication for reconfigurable platforms. In *CHES*, volume 4249, pages 311–323. Springer, 2006.

[23] Christian Wachsmann, Liqun Chen, Kurt Dietrich, Hans Löhr, Ahmad-Reza Sadeghi, and Johannes Winter. Lightweight anonymous authentication with TLS and DAA for embedded mobile devices. In *Information Security*, pages 84–98. Springer, 2010.

[24] Li Xi, Dengguo Feng, Yu Qin, Feng Wei, Jianxiong Shao, and Bo Yang. Direct anonymous attestation in practice: Implementation and efficient revocation. In *Privacy, Security and Trust (PST), 2014 Twelfth Annual International Conference on*, pages 67–74. IEEE, 2014.

[25] Bo Yang, Dengguo Feng, and Yu Qin. A lightweight anonymous mobile shopping scheme based on DAA for trusted mobile platform. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on*, pages 9–17. IEEE, 2014.

[26] Bo Yang, Kang Yang, Yu Qin, Zhenfeng Zhang, and Dengguo Feng. DAA-TZ: An efficient DAA scheme for mobile devices using ARM trustzone. In *Trust and Trustworthy Computing*, pages 209–227. Springer, 2015.

[27] Jiliang Zhang, Yaping Lin, Yongqiang Lyu, and Gang Qu. A PUF-FSM binding scheme for FPGA IP protection and pay-per-device licensing. *Information Forensics and Security, IEEE Transactions on*, 10(6):1137–1150, 2015.

[28] Qianying Zhang, Shijun Zhao, Li Xi, Wei Feng, and Dengguo Feng. Mdaak: A flexible and efficient framework for direct anonymous attestation on mobile devices. In *Information and Communications Security*, pages 31–48. Springer, 2014.