Intel Corporate Research Council



**Intel Corporate Research Council/ University Research Office (URO)**

**GRANT PROPOSAL COVERSHEET & PRIVACY POLICY NOTICE**

Intel is committed to respecting your privacy. The information you provide will be used and retained for processing and funding your grant/gift, for Intel’s audit purposes, and for grant-related correspondence. For more information regarding Intel’s personal information handling practices, please visit [www.Intel.com/Privacy.](http://www.intel.com/Privacy)

 **(For grants other than conference sponsorships)** Check here to certify that youare not



engaged in prior agreements (e.g. government contracts) that constraint your ability to negotiate mutually agreeable intellectual property terms for this grant

 **(For grants other than conference sponsorships)** Check here to notify us that youare



engaged in prior agreements (e.g. government contracts) that constrain your ability to negotiate mutually agreeable intellectual property terms for this grant

 Check here to opt in for occasional URO updates - announcements or newsletters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Proposal Title:** | | | | |
| **Key-Aggregate Bit-Stream Encryption for Multi-Tenant FPGA Platforms** | | | | |
| **University Name / Receiving Organization** | |  | **Department/Discipline** | |
| **Indian Institute of Technology Kharagpur, India** | | | **Department of Computer Science and Engineering** | |
| **Representative Authorized** | |  | **Principal Investigator Information** | |
| **To Conduct Grant Administration** | |  |  | |
|  |  |  |
| **Contact Name** | Dr. Pallab Dasgupta |  | **PI Name** | Dr. Debdeep Mukhopadhyay |
|  |  |
| **Mailing** | Dean, Sponsored Research and Industrial Consultancy (SRIC)  IIT Kharagpur,  Kharagpur-721302,  West Bengal, India | | **Mailing Address** | Department of CSE,  IIT Kharagpur,  Kharagpur-721302,  West Bengal, India |
| **Address** |  |
|  |  |
| **Phone #** | +91-3222-282037 |  | **Phone #** | +91-3222-282352 |
| **Fax #** |  |  | **Fax #** | +91-3222-277190 |
| **E-Mail** | [deansr@hijli.iitkgp.ernet.in](mailto:deansr@hijli.iitkgp.ernet.in) |  | **E-Mail Address** | [debdeep@cse.iitkgp.ernet.in](mailto:debdeep@cse.iitkgp.ernet.in) |
| **Address** | [pallab@cse.iitkgp.ernet.in](mailto:pallab@cse.iitkgp.ernet.in) |  |  | [debdeep.mukhopadhyay@gmail.com](mailto:debdeep.mukhopadhyay@gmail.com) |
|  |  |  | **Project /PI URL** | cse.iitkgp.ac.in/~debdeep |
|  |  |  | **Co-Investigator/Students** | Arnab Bag (PhD Candidate)  Sikhar Patranabis (PhD Candidate) |
| **Amount of** |  |  |  |  |
| **Cash** |  |  |  |  |
| **Requested** |  |  |  |  |
|  | FPGA-based applications in the automotive and cloud-based settings have given rise to IP protection requirements for a multi-tenant usage model, where multiple mutually distrusting parties co-exist on the same FPGA. Existing IP protection techniques based on bit-stream encryption/authentication do not address certain security issues specific to multi-tenant usage models, such as bit-stream isolation and collusion resistance.We propose “Key-Aggregate Bit-Stream Encryption” - a cryptographically secure mechanism for secure multi-tenant sharing of FPGA resources. Our proposal is novel in the sense that it allows bit-streams from different tenants to be encrypted with separate keys, and efficiently manages the key storage overhead by aggregating the access rights to any arbitrary number of encrypted bit-streams into a single constant-size aggregate key. Our proposal uses standard crypto-primitives such as block ciphers and elliptic-curve based bilinear pairings, and is secure under standard hardness assumptions. Additionally, it is efficiently implementable on a hardware-software co-processor platform. | | | |
| **Additional Comments:** |
|  |

**Key-Aggregate Bit-Stream Encryption for Multi-Tenant FPGA Platforms**

### **Debdeep Mukhopadhyay, Arnab Bag, Sikhar Patranabis**

### Department of Computer Science and Engineering

### Indian Institute of Technology Kharagpur, India

[debdeep@cse.iitkgp.ernet.in](mailto:debdeep@cse.iitkgp.ernet.in), [arnabbag@ iitkgp.ac.in](mailto:arnabbag@%20iitkgp.ac.in), [sikhar.patranabis@iitkgp.ac.in](mailto:sikhar.patranabis@iitkgp.ac.in)

Detailed Technical Proposal (RV1: FPGA Security)

**Background and Motivation.** The advent of cloud computing today offers distributed platform for storage and analytics on large volumes of data, albeit at the risk of security threats arising from malicious service providers and external adversaries. In particular, with respect to FPGA-based applications, this has heralded a paradigm shift from IP protection for designs originating from single vendor, to a multi-tenant usage model, where multiple mutually distrusting parties could be running their individual applications on the same FPGA platform.

**Problem Statement.** In this proposal, we address the following open problems, outlined in research vector RV1:

* Enumerating the security threats deriving from the multi-tenant usage model in the presence of mutually distrusting tenants and platform owners. This is particularly relevant for applications in the automotive setting and cloud-based deployment setting.
* Proposing a cryptographically secure mechanism for secure multi-tenant sharing of FPGA resources in the presence of malicious adversaries

**Security Threats in the Multi-tenant FPGA Model.** We formally define a multi-tenant FPGA model and enumerate the security challenges thereof. Suppose there are mutually distrusting parties that wish to allow their applications to be run on an FPGA platform. The platform would typically comprise of multiple FPGAs that may be in turn be licensed out for use by single/multiple clients. The bit-streams corresponding to these applications are hosted in an untrusted environment, such as a third-party provisioned cloud. Depending on the client-license, an FPGA should be able to run a specific subset of these applications, such that. For example, if a client has the license for applications from party- and party- (where), then the corresponding FPGA should only be able to run these applications, and nothing else. With this framework in mind, we enumerate the resulting security challenges that FPGA vendors must cater to:

1. **Bit-Stream Isolation:** In a multi-tenant environment comprising of mutually distrusting parties, it is important to ensure that ensuring access to IP from one party, say party- does not inherently compromise the IP security of another party, say party-.
2. **Collusion-Resistance:** Multiple clients with their own individual licenses should not be able to collude and achieve unauthorized access to any set of bit-stream files not covered by their licenses. For example, suppose Client-1 has a license for applications from parties 1, 3 and 5, while Client-2 has a license for applications from parties 2, 4 and 6. However, they should not be able to collude and use mutually assimilated information to expose a bit-stream file from, say party-7, which none of them are originally authorized to access. Collusion-resistance is a practical security requirement in any multi-client application.
3. **Access Revocation:** It must be possible to efficiently revoke the access of a client to any application hosted by a certain party upon expiry of the appropriate license. In other words, at any point of time, the client should only be able to access the subset of applications for which she has a valid license, irrespective of her access history.

**A Naïve Solution.** A straightforward solution to the aforementioned problem could be as follows: each party encrypts its respective bit-files with its own key, which is then securely transmitted to the FPGA vendor via a secure channel. At any time, based on the license possessed by a client, the corresponding FPGA will use only the keys corresponding to the subset of applications that a client has access to. This approach, when appropriately strengthened using authentication mechanisms, works well in a single-tenant scenario. However, as the number of parties grows, this approach imposes a significantly high key-management overhead, involving both secure key-storage and secure-channel based key transmission, the complexity of which grows linearly with. This seriously limits its scalability, and motivates the need for more efficient solutions.

**Our Proposition.** We propose a novel primitive - key-aggregate bit-stream encryption (KABE) to addresses the above requirement. KABE is a public-key system with a single *master secret key.* In order to ensure data isolation and collusion resistance in a multi-tenant scenario, the following two-level strategy may be adopted:

* The first level uses the naïve strategy described above, where each party uses its own individual key to encrypt the bit-streams corresponding to its applications, and places these encrypted bit-streams in a shared data environment such as the cloud. The encryption algorithm used could be any standard symmetric-key block cipher algorithms such as AES-128, which provides good encryption performance for bulk data. Since each party uses its own key to encrypt its bit-stream files, data isolation and data privacy are automatically guaranteed. However, the resulting system now has distinct keys, say to. The challenge is to manage such a large number of keys efficiently when provisioning the same to the FPGA vendor. This issue of efficient key-management is addressed by KABE at the second level, as discussed next.
* At the second level, each of these keys, to, are further encrypted by the respective parties, and sent to the FPGA vendor, who may embed these encrypted keys into each FPGA. This second-level encryption would typically involve bilinear pairings on elliptic curves. Since the keys are encrypted, there is no need to transmit them to the vendor via a secure channel. Now suppose a client purchases a license to access a subset of the encrypted bit-streams. The FPGA vendor issues a single constant-size *aggregate key*  which can be used to recover the first-level keys corresponding to the subset. On the other hand, cannot be used to gain unauthorized access to any keys outside the subset. This property plays a major role in ensuring collusion resistance. The client can first use this aggregate key to recover the corresponding secret keys used in the first level, and perform a second decryption to recover the respective bit-streams. Thus the second level provides efficient key management by reducing the key transmission requirement from separate keys to a single aggregate key with size independent of.
* A final challenge is in transmitting the aggregate key securely to multiple clients. To tackle this issue, KABE may be combined with *broadcast encryption* - a well-known cryptographic primitive that securely disseminates the aggregate key among any number of clients without the need for explicit secure channels.

**Security Guarantees.** Our constructions for KABE satisfy standard notions of cryptographic security such as data privacy, data isolation and collusion-resistance under well-known hardness assumptions. The constructions are based on elliptic curve groups and use bilinear pairings, for which several efficient implementations in both hardware and software across multiple platforms have been proposed in the literature.

**Comparison with State-of-the-Art IP Protection Techniques.** Existing IP protection techniques implemented by FPGA vendors can be broadly categorized into two categories, each of which inherently assume that the implemented design originates from a single entity, and are hence not directly applicable to usage models where multiple, mutually distrusting entities co-exist on a single FPGA.

* **FPGAs with Built-In Encryption Abilities.** The first approach is to use FPGAs with built-in encryption capabilities to prevent configuration bit-streams from being cloned or copied [4, 5, 9, 10, 11]. This is typically prevalent in newer families of SRAM FPGAs that require an external configuration memory. The configuration bit-stream is typically encrypted using a secret key and stored in the external memory. On power-up, the encrypted bit-stream is read from the configuration memory into the FPGA, decrypted and loaded into the fabric. While the key-management overhead in this technique is manageable when the IP originates from a single entity, it tends to suffer from potentially exponential blowups for key storage and management in a multi-tenant scenario, since bit-streams obtained from different sources must be encrypted with different keys. Our proposed methodology, on the other hand, avoids such a blowup by aggregating the access rights to multiple bit-streams from different parties into a single aggregate key with constant overhead. This unique feature makes our technique suitable for multi-tenant scenarios.
* **FPGA Intrinsic Physically Unclonable Functions.** The second approach is to use FPGA intrinsic physically unclonable functions (PUFs) based on SRAM memory randomness [3, 6, 7, 8]. The idea is to embed each FPGA with a PUF instance, with specific challenge-response characteristics that are used for authentication prior to bit-stream access. While such techniques avoid key storage requirements and can be used to protect both single FPGA configurations and third-party FPGA IP cores, their security guarantee are based on strong uniqueness and stability requirements on FPGA-intrinsic PUFs, which is sometimes difficult to achieve in practice. In addition, they are not typically designed to handle scenarios where mutually distrusting parties co-exist on the *same* FPGA, which would require that multiple PUF instances (a unique instance per party) be implemented on the same FPGA. The practicality of achieving the same, as well the resulting security implications, are not immediately apparent from existing studies.

**Demonstration and Experimentation Plan.** We intend to develop a prototype end-to-end tool for our proposed KABE scheme. The implementation would typically be based a software-hardware coprocessor based architecture. Since KABE targets IP protection for FPGA platforms, we obviously target FPGA platforms for our hardware implementations. Modern FPGAs offer high flexibility and rapid prototyping capabilities in the face of increased time-to-market concerns, while also reducing incremental costs for design changes and minimizing reliability concerns. We enumerate below the chief architectural components of the proposed KABE implementation, along with a proposed hardware/software split:

* **Overall Protocol Framework.** Our prototype tool would include three main types of parties - the mutually distrusting tenants, the FPGA vendors and the clients. The following sub-components of the overall protocol framework are proposed to be implemented in software:
  + Inter-communication between the aforementioned parties
  + Transmission and storage of encrypted bit-streams and the corresponding encryption keys used by the tenants
  + Generation of client licenses and transmission of the corresponding aggregate keys from the vendor to the client
* **Basic Cryptographic Primitives.** The cryptographic core of our implementation would typically use block ciphers such as AES-128 for bulk encryption of bit-streams (Level-1 of protocol), and bilinear pairings on elliptic curves for efficient key-management and aggregation capabilities (Level-2 of protocol). We propose implementing dedicated hardware accelerators for each of these cryptographic primitives, which are already extensively documented in the cryptographic literature. A possible candidate choice is the Barreto-Naehrig (BN) family of elliptic curves, which give rise to area-optimal bilinear pairing implementations.

Figure (TBD)

Statement of Work, Schedule and Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Work Statement | Deliverables | Time Frame |
| Phase-0 | * Groundwork and understanding of elliptic curve cryptosystems and bilinear pairings * Identification of security concerns to be addressed in the multi-tenant FPGA setting * Development of a simple pairing-based cryptosystem with key-aggregation features that addresses such security concerns | * A **key-aggregate cryptosystem** that is provably data private against chosen-ciphertext attacks as well as multi-party collusion attacks * Proof of Concept software implementation of the key-aggregate cryptosystem | Completed  ( references [1,2]) |
| Phase-1 | * Development of an end-to-end architecture for the proposed two-level KABE protocol * Identification of the appropriate target platform (preferably a hardware-software coprocessor) for the prototype tool development | * Prototype end-to-end architecture description for the KABE protocol * Broad overview of the necessary cryptographic, communication and storage modules in the design * Hardware-Software partitioning of the overall design as per target platform | 8 months |
| Phase-2 | * Specifying the choice of cryptographic cores to be used in the implementation e.g. elliptic curve libraries, block cipher cores, and the security guarantees thereof * Development of dedicated hardware accelerators for the chosen cryptographic cores | * RTL description of block cipher cores in Verilog/VHDL * RTL description of elliptic curve cores in Verilog/VHDL * Summary of Area/Timing requirements for each cryptographic core along with efficiency assessment | 8 months |
| Phase-3 | * Specifying the choice of auxiliary modules for communication and synchronization between the various parties in the overall protocol design * Identification of appropriate storage mechanisms (for instance, databases such as MongoDB supporting efficient read/write/retrieval operations), for the ciphertexts generated by the protocol | * Software implementation of the communication modules in the design * Database set up and synchronized with the communication modules for efficient storage of ciphertexts | 8 months |
| Phase-4 | * Integration of hardware and software modules * Deployment on the target platform | * First fully functional version of prototype tool on the target platform | 6 months |
| Phase-5 | * Testing and Validation | * Updated and tested version of prototype tool on the target platform | 6 months |

Cost Summary

Expected Resources, Expenses, Overhead, Equipment in USD (TBD)

Overview of Proposal Team

Brief Biographies of Proposal Team

|  |  |
| --- | --- |
| **Dr. Debdeep Mukhopadhyay**  **(Project PI)** | Dr. Debdeep Mukhopadhyay is currently an Associate Professor at the Department of Computer Science and Engineering, Indian Institute of Technology at Kharagpur, India and a visiting scientist at School of Computer Science and Engineering at NTU, Singapore. At IIT Kharagpur he initiated the Secured Embedded Architecture Laboratory (SEAL), with a focus on Embedded Security and Side Channel Attacks (http://cse.iitkgp.ac.in/resgrp/seal/). Prior to this he worked as a visiting Associate Professor of NYU-Shanghai. He had also served as an Assistant Professor at IIT Madras, India and as a Visiting Researcher at NYU Polytechnic School of Engineering under the Indo-US STF Fellowship. He holds a PhD, an MS, and a B. Tech from IIT Kharagpur, India. Dr. Mukhopadhyay’s research interests are Cryptography, Hardware Security, and VLSI. His books include Cryptography and Network Security (Mc Graw Hills), Hardware Security: Design, Threats, and Safeguards (CRC Press), and Timing Channels in Cryptography (Springer). He has written more than 100 papers in peer-reviewed conferences and journals and has collaborated with several Indian and Foreign Organizations. Dr. Mukhopadhyay is the recipient of the prestigious Swarnajayanti DST Fellowship 2015-16, Young Scientist award from the Indian National Science Academy, the Young Engineer award from the Indian National Academy of Engineers, and is a Young Associate of the Indian Academy of Science. He was also awarded the Outstanding Young Faculty fellowship in 2011 from IIT Kharagpur, and the Techno-Inventor Best PhD award by the Indian Semiconductor Association. He has recently incubated a start-up on Hardware Security, ESP Pvt Ltd at IIT Kharagpur (http://esp-research.com/). |
| **Arnab Bag**  **(Full-Time PhD Candidate)** | Arnab Bag is currently a PhD candidate under the supervision of Dr. Debdeep Mukhopadhyay at the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India. He received his B.Tech and M.Tech degrees under the dual degree program from the Department of Electronics and Electrical Communication Engineering, IIT Kharagpur, in 2017. His current research interests include FPGA-based architectural designs for cryptographic primitives, automotive security, and hardware security. |
| **Sikhar Patranabis**  **(Full-Time PhD Candidate)** | Sikhar Patranabis is currently a PhD candidate under the supervision of Dr. Debdeep Mukhopadhyay at the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India. He received his B. Tech degree from the same department in 2015, and was the President’s Gold Medal awardee for securing the highest grade point average among all graduating students of his batch. He is currently an IBM PhD Fellow, and has completed internships with IBM IRL and NTU, Singapore. His current research interests include public-key cryptosystems for cloud-based applications, lightweight cryptography and hardware security. He has published his research work in reputed journals such as IEEE Transactions on Computers and IEEE Transactions on Information Forensics and Security. Sikhar is also actively involved in a project titled “LightCrypto: Ultra-Light-weight Robust Crypto-Architectures for Performance and Energy” sponsored by Intel Labs, Hillsborough, Oregon, USA, since May, 2015. |

References

1. Sikhar Patranabis, Yash Shrivastava, Debdeep Mukhopadhyay: Provably Secure Key-Aggregate Cryptosystems with Broadcast Aggregate Keys for Online Data Sharing on the Cloud. IEEE Transactions on Computers, 2017
2. Sikhar Patranabis, Yash Shrivastava, Debdeep Mukhopadhyay: Dynamic Key-Aggregate Cryptosystem on Elliptic Curves for Online Data Sharing. INDOCRYPT 2015
3. Aydin Aysu, Patrick Schaumont: Hardware/software co-design of physical unclonable function based authentications on FPGAs. Microprocessors and Microsystems - Embedded Hardware Design, 2015
4. Amir Moradi, Alessandro Barenghi, Timo Kasper, Christof Paar: On the vulnerability of FPGA bitstream encryption against power analysis attacks: extracting keys from xilinx Virtex-II FPGAs. Proceedings of the 18th ACM conference on Computer and communications security. ACM, 2011.
5. Altera White Paper: Protecting the FPGA Design From Common Threats. 2009 (<https://www.altera.com/en_US/pdfs/literature/wp/wp-01111-anti-tamper.pdf>)
6. Jorge Guajardo, Sandeep S. Kumar, Geert Jan Schrijen, Pim Tuyls: Brand and IP protection with physical unclonable functions. ISCAS 2008
7. Sandeep S. Kumar, Jorge Guajardo, Roel Maes, Geert Jan Schrijen, Pim Tuyls: The Butterfly PUF: Protecting IP on every FPGA. HOST 2008
8. Jorge Guajardo, Sandeep S. Kumar, Geert Jan Schrijen, Pim Tuyls: FPGA Intrinsic PUFs and Their Use for IP Protection. CHES 2007
9. Eric Simpson, Patrick Schaumont: Offline hardware/software authentication for reconfigurable platforms. CHES 2006.
10. Tom Kean: Cryptographic rights management of FPGA intellectual property cores. Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays. ACM, 2002.
11. KW Yip, TS Ng: Partial-encryption technique for intellectual property protection of FPGA-based products. IEEE Transactions on Consumer Electronics, 2000