POSTER: SHAMROCK: Self Contained Cryptography and Key Management Processor

Daniil Utin
MIT Lincoln Laboratory
244 Wood Street
Lexington, 02420, USA
danu@ll.mit.edu

Roger Khazan
MIT Lincoln Laboratory
244 Wood Street
Lexington, 02420, USA
rkh@ll.mit.edu

Joshua Kramer
MIT Lincoln Laboratory
244 Wood Street
Lexington, 02420, USA
joshua.kramer@ll.mit.edu

Michael Vai
MIT Lincoln Laboratory
244 Wood Street
Lexington, 02420, USA
mvai@ll.mit.edu

David Whelihan
MIT Lincoln Laboratory
244 Wood Street
Lexington, 02420, USA
david.whelihan@ll.mit.edu

ABSTRACT
In this poster, we describe a one-size-fits-many Intellectual Property (IP) core which integrates advanced key management technology and streaming encryption into a single component to protect data in-transit.

Categories and Subject Descriptors
E.3 [**Data Encryption**]: Public key cryptosystems; Standards

Keywords
key management; cyber security; cryptography; self-contained encryption; high-Assurance data protection; identity management; hardware IP core; FPGA; cryptographic component; key management protocol.

PROBLEM
Modern cryptography offers a variety of schemes for the protection of information at-rest on devices and in-transit among devices. A cryptographic scheme typically "scrambles" or "unscrambles" information using a data-permutation algorithm and a short cryptographic key. The security of the scheme depends on the properties of the algorithm and the quality and secrecy of the key. Thus, cryptographic keys need to be created and managed carefully. In particular, they need to be protected at-rest and in-transit, which itself calls for the use of various cryptographic schemes.

Although many cryptographic schemes have been standardized and implemented efficiently in software and hardware, these solutions are not universally used or embedded in devices. The two main reasons are the lack of generic, easy-to-deploy, and easy-to-use solutions for key management, and the challenge of integrating various cryptographic and key management components into a holistically secure design. While individual cryptographic components exist and may be known to be secure, there is no known recipe for integrating different components into secure designs that guarantee security of keys and other information, at-rest and in-transit. It is in such integration that major challenges exist and vulnerabilities are oftentimes introduced.

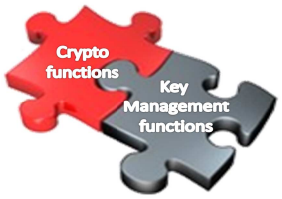
SOLUTION
MIT Lincoln Laboratory developed a one-size-fits-many Intellectual Property (IP) core, called SHAMROCK[1]. SHAMROCK integrates advanced key management technology and streaming encryption into a single component to protect data in-transit. It is designed for low Size, Weight, and Power applications and can be embedded into a wide variety of devices, enabling entire systems to be secured with the same component.

This work is sponsored by the United States Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

CCS'13, November 4–8, 2013, Berlin, Germany.
ACM 978-1-4503-2477-9/13/11.
http://dx.doi.org/10.1145/2508859.2512515

[1] SHAMROCK stands for Self-contained High-Assurance MicRO Cryptography and Key-management.

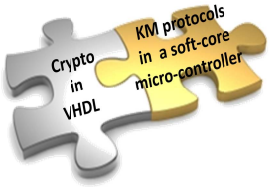
In this poster, we describe the SHAMROCK embedment, key-management functionality, upgradeability and performance in terms of energy and speed.



Using a simple, accessible interface, a device containing SHAMROCK chip can use the SHAMROCK's cryptographic components, such as an Advanced Encryption Standard (AES) cipher core, to secure its data internally and to protect its communication with other devices [1]. The device can also use SHAMROCK to handle all of the key management tasks required for the operation of the cryptographic components.

These key management tasks may include operations such as creating cryptographic keys, associating keys with their purposes, protecting keys at rest in both volatile and non-volatile memory, making keys available for authorized encryptions and authorized decryptions, delivering keys securely to authorized remote devices, archiving keys, evolving keys with time, retiring keys, etc. SHAMROCK integrates all of the components into a self-contained, secure design thereby simplifying the task of incorporating cryptographic protections into applications.

While a self-contained, integrated crypto and key management solution is novel in itself, we have made a number of inventions in creating this integrated design. One of the innovations is a way to attain high performance, low power, flexibility, and extensibility – all at the same time. This is achieved by implementing certain, typically computationally demanding components, such as the standard cryptographic functions, directly in hardware (for example using VHDL [2]), while other components, such as key management protocols, in a higher-level language, such as C, in a softcore micro-controller inside the integrated circuit. The soft-core micro-controller, allows for different key management protocols and their extensions to be easily created for and deployed in SHAMROCK, even after the integrated circuit core is manufactured and integrated into an application. The hardware implementations of the cryptographic components used by the application and by the key management components facilitate lower-power, higher-speed operations. Other innovations strengthen security through hardware design and reduce the effort and cost of verifying and modifying the design.



IMPLEMENTATION AND PERFORMANCE ANALYSIS

Implementation and performance analysis: SHAMROCK has been realized as an IP core on two types of Xilinx FPGAs: Virtex-6 and Kintex-7 [3]. The key management protocols are implemented in C programming language and execute in the Tensilica 106micro soft-core micro-controller [4]. 106micro is a central micro-controller and is a part of the SHAMROCK IP core. It manages and coordinates a number of hardware-accelerated cryptographic functions implemented in VHDL. Kintex-7 is the latest generation of XILINX's FPGAs, with the static power consumption being much lower (more than 50%) than that of Virtex-6.

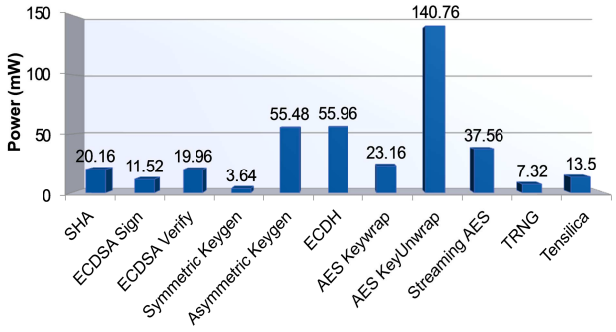


Figure 1. Kintex 7 Dynamic Power (50MHz)

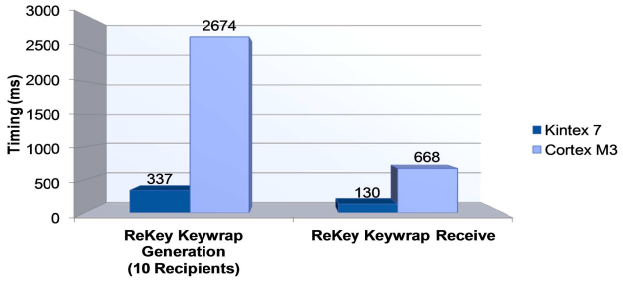


Figure 2. Key Management Function Timing (FPGA@50MHz / CPU@72MHz)

Figure 1 shows the *dynamic* power consumption on Kintex-7 of the different cryptographic operations in SHAMROCK. Typically, most of these operations would run infrequently, except for Streaming AES, which protects application data. Comparing the two figures, observe that dynamic power is significantly less than static power, especially for Streaming AES. Key management functions, such as the generation and processing of key wraps, involve a whole set of cryptographic operations. Figure 2 depicts the time to perform these two operations on Kintex-7 and compares them to those on an ARM Cortex M3 microprocessor [5]. The former is close to an order of magnitude faster than the ARM CPU.

References

[1] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[2] https://en.wikipedia.org/wiki/VHDL

[3] https://en.wikipedia.org/wiki/Xilinx

[4] https://en.wikipedia.org/wiki/Tensilica

[5] https://en.wikipedia.org/wiki/ARM_Cortex-M