In Proceedings of: IEEE Symposium on Field-Programmable Custom
Computing Machines (FCCM), Napa, CA, April 20-23, 2004

Secure Remote Control of Field-programmable Network Devices

Haoyu Song, Jing Lu, John Lockwood, James Moscola
Washington University in St. Louis, USA
http://www.arl.wustl.edu/projects/fpx/reconfig.htm

Abstract

A circuit and an associated lightweight protocol have been developed to secure communication between a control console and remote programmable network devices[1]*. The circuit provides encryption, data integrity checking and sequence number verification to ensure confidentiality, integrity and authentication of control messages sent over the public Internet. All of these functions are performed directly in FPGA hardware to provide high throughput and near-zero latency. The circuit has been used to control and configure remote firewalls and intrusion detection systems. The circuit could also be used to control and configure other distributed network applications.*

1. Introduction

New types of distributed firewalls [4], extensible network routers, Intrusion Detection and Prevention Systems (IDPS), and Internet-enabled sensors use reconfigurable hardware devices because they offer both high performance and flexibility. In order to distribute devices over a large geographic area, robust security mechanisms are needed to protect the network devices from unauthorized access and to ensure the integrity of control messages sent over the public Internet.

Existing software-based security frameworks require a large computational effort to encrypt and decrypt data and can bottleneck system performance. We have developed a lightweight solution to the secure control of network devices that uses only hardware mechanisms to send and receive control messages. It has been applied to both a firewall and an IDPS, which are built on the Field-programmable Port Extender (FPX) [3] platform. New features can be added into the system through the secure control channel whenever necessary.

2 Architecture and Protocol

A circuit is built that allows communications between a control console and Remote Network Devices (RND) over the Internet through encrypted and authenticated control packets. A functional module called Secure Remote Control Engine (SRCE) performs these tasks to ensure that only authorized access to the RND is allowed. Figure 1 shows a typical configuration of the secure remote control of network devices. The SRCE can be an embedded component of the RND. Because there is no software involved in network processing, the devices cannot be hacked by any of the vulnerabilities found in existing operating systems.

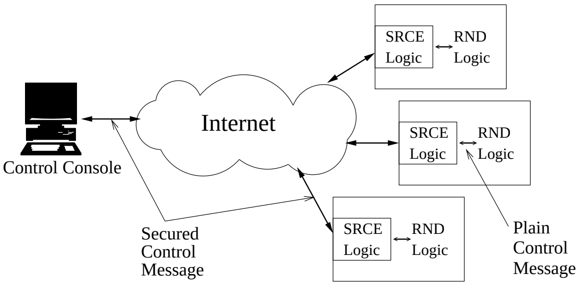

Figure 1. Secure Remote Reconfiguration

Encrypted control messages are encapsulated in UDP packets. SRCE logic identifies control packets and passes them to the decryption core shown at the left of Figure 2. A secret key is shared by both sender and receiver. Encryption alone is not sufficient to make the RND control secure. To maintain data integrity, a Cyclic Redundancy Check (CRC) is also generated for each control message. To defend against replay attacks, each packet is also assigned a unique Sequence Number (SN). The use of all mechanisms described above, attackers can not generate bogus packets to compromise the network devices or obtain the knowledge of the communication content.

Figure 2 illustrates the logic components in the SRCE. When control packets that match the device's IP address and the UDP port number arrive, they are sent to the decryption core and integrity check circuits. Valid control packets

[1] This work was supported by a grant from Global Velocity. The authors of this paper have received equity from the the license of technology to that company. John Lockwood has served as a consultant and co-founder to Global Velocity.

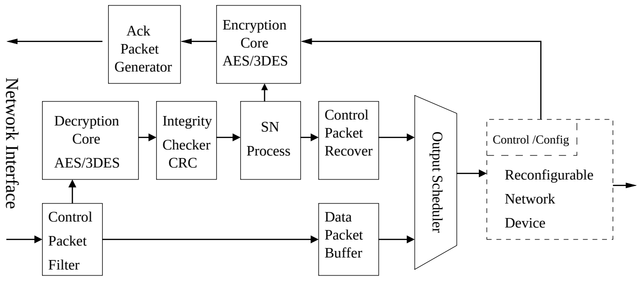

Figure 2. SRCE Architecture

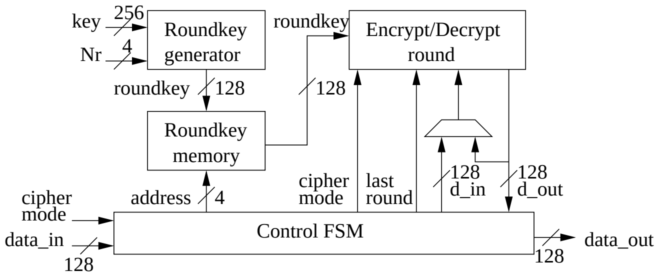

Figure 3. AES Hardware Implementation

are forwarded to the following block to verify the SN and control packet flags. If a control packet is identified as genuine and in-order, it will be used to control or configure the RND. At the same time, an acknowledgement packet with next expected SN is generated, encrypted and sent back to the control console.

The SRCE establishes, maintains and terminates the secure connection. By using the Go-back-N ARQ, a security protocol with flow control and error control is designed to secure the communications between the control console and the RND. A sliding window mechanism provides flow control. Communication is divided into two phases: the session establishment phase and configuration phase. we implement the batch retransmission strategy because of the requirement that the packets must configure or reprogramme the network device in order. For each new connection, a random SN is generated in the SRCE and sent back to the control console. The SN is incremented as control messages pass through the connection. The lightweight protocol supports end-to-end communication. Bogus control packets are dropped by applying encryption and authentication to the control packets. Replay attacks are prevented by using randomly generated SN. The control console only accepts authenticated positive acknowledgement and retransmit the lost packets based on a time-out mechanism. Whenever the RND is under attack or network fails, the control console can be aware of these problems and take action immediately.

3 Applications

We implement the SRCE for both a hardware firewall [5] and an IDPS [2]. Control packets sent to and from the protected system are encrypted using AES (Rijndael) [1] algorithm. Figure 3 illustrates our hardware implementation of AES. The AES encryption core supports keys of 128, 192 or 256 bits. Sixteen 256-byte substitution boxes (Sboxes) are used for each encryption round and another sixteen 256-byte inverse Sboxes for each decryption. To save chip resources, we implement Sboxes in block RAMs, so that each AES round takes two clock cycles. An iterative architecture was adopted to lower the chip utilization. A throughput of 547.8 Mbps was achieved on Xilinx Virtex2000E-6 device for a 128-bit key. With a loop-unrolling architecture, 10x speedup could be achieved on a larger device.

Through the SRCE, the systems can be securely configured from remote sites. To implement the system, one FPX card is programmed to implement the SRCE circuit. This FPX card is placed in front of another FPX card that implements the Firewall or IDPS.

Testing is done in a live network. NCHARGE [6] is used to generate control packets. The web interface is augmented with a selection box to choose AES key length. The web interface calls a CGI script to interpret the information, encrypt it, assemble and send the packets out via a Gigabit Link to the tested system. The NCHARGE then reads acknowledgement packets from the link. Any Acknowledgement received is first decrypted and then determined if a retransmission is needed. The synthesized module supports up to 2Gbps overall throughput.

References

[1] FIPS - 197 : Advanced Encryption Standard (AES). Online:`http://csrc.nist.gov/encryption/aes/`, Feb. 2001.

[2] M. Attig, S. Dharmapurikar, and J. Lockwood. Implementation of bloom filters for string matching. In *FCCM 2004*.

[3] J. Lockwood. An open platform for development of network processing modules in reprogrammable hardware. In *IEC DesignCon'01*.

[4] J. Lockwood, J. Moscola, and M. Kulig. Internet worm and virus protection in dynamically reconfigurable hardware. In *MAPLD 2003*.

[5] J. Lockwood, C. Neely, and C. Zuver. An extensible, system-on-programmable-chip, content-aware Internet firewall. In *FPL 2003*.

[6] T. Sproull, J. Lockwood, and D. Taylor. Control and Configuration Software for a Reconfigurable Networking Hardware Platform. In *FCCM 2002*.