Memory Protection with Dynamic Authentication Trees

Matthew Millar, Marcin Łukowiak
Department of Computer Engineering
Rochester Institute of Technology
mxm1898@rit.edu, mxleec@rit.edu

Stanisław Radziszowski
Department of Computer Science
Rochester Institute of Technology
spr@cs.rit.edu

Extended Abstract

Embedded devices that process sensitive data and provide essential services have become increasingly common as modern digital infrastructures have grown at a rapid pace. This raises security concerns as there are more reasons than ever for a malicious party to try to exploit these systems. In an attempt to provide security, encryption methods are oftentimes applied to any sensitive data to provide confidentiality. However, data confidentiality itself is not enough to fully protect a system from an attacker. For example, erroneous data may be injected into the system in an attempt to disrupt the normal functionality of the device. In order to protect against such attacks, it is necessary to verify that any data the device processes is provided by the expected source. In addition, there needs to be a method of ensuring that the data has not been tampered with. Methods of authentication are then used to confirm the integrity of data processed in the system. Existing authentication methods, such as hashes and message authentication codes (MACs), are able to provide the intended protection; however, some of these methods are costly in terms of the device resources and performance overhead required to implement them. In this paper we present a new method of dynamic authentication trees, which update a tree structure based on a processor's memory access patterns. The key features of the approach are as follows:

- *Authentication:* Modified block-level added redundancy explicit authentication (block-level AREA) scheme is employed in our ordered Dynamic Authentication Tree (DAT) method. Utilizing this approach has the benefit of providing encryption inherently with the authentication operations.
- *Tree Structure:* The structure of the tree depends on the weighted frequency of accesses to each data node.
- *Tree Nodes:* The leaf nodes of the authentication tree structure contain the data blocks that are directly protected by the tree. These nodes are referred to as data nodes, and are separated into two parts: the data that is being protected and the nonce. The nonce contains the tree metadata required for tree traversal and a count of the number of times the node has been accessed.
- *FPGA Design:* An AXI-4 based framework is developed as a transparent and highly customizable memory controller. This design is then synthesized onto an FPGA and verified.
- While this design is motivated by vulnerabilities in embedded systems, the method presented is scalable and may be applied to any computing system. Different configuration options are provided allowing for the design to be used in applications with different restraints and requirements.

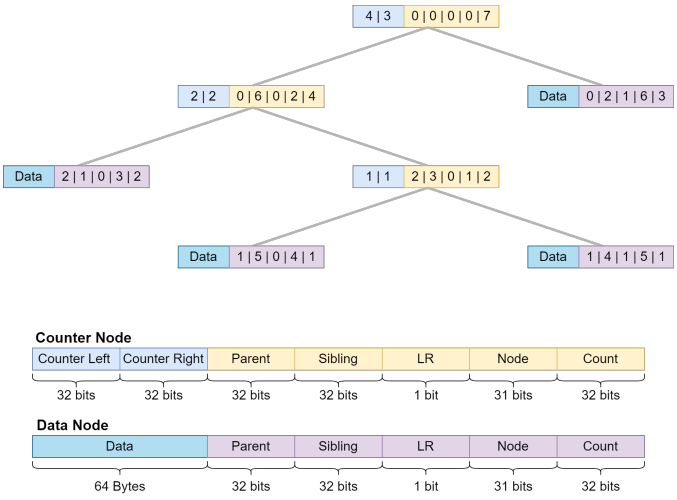

Fig. 1. Ordered Dynamic Authentication Tree