SANDIA REPORT
SAND2012-5041
Unlimited Release
Printed June 2012

Authenticating Cache

Tyler B. Smith and Jorge M. Urrea

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2012-5041
Unlimited Release
Printed June 2012

Authenticating Cache

Tyler B. Smith
Surety Electronics and Software

Jorge M. Urrea
Critical Infrastructure Systems

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0451

Abstract

The aim of the Authenticating Cache architecture is to ensure that machine instructions in a Read Only Memory (ROM) are legitimate from the time the ROM image is signed (immediately after compilation) to the time they are placed in the cache for the processor to consume. The proposed architecture allows the detection of ROM image modifications during distribution or when it is loaded into memory. It also ensures that modified instructions will not execute in the processor—as the cache will not be loaded with a page that fails an integrity check. The authenticity of the instruction stream can also be verified in this architecture. The combination of integrity and authenticity assurance greatly improves the security profile of a system.

AUTHENTICATING CACHE

The aim of the Authenticating Cache architecture is to ensure that machine instructions in a Read Only Memory (ROM) are legitimate from the time the ROM image is signed (immediately after compilation) to the time they are placed in the cache for the processor to consume. The proposed architecture allows the detection of ROM image modifications during distribution or when it is loaded into memory. It also ensures that modified instructions will not execute in the processor—as the cache will not be loaded with a page that fails a signature check (integrity check).

In addition to the integrity checks mentioned above, the authenticity of the sender could be verified by keeping the hashing function's Initialization Vector (IV) a shared secret between trusted parties. If an adversary attempted to provide a false ROM, s/he would not be able to calculate acceptable hash results because s/he would not know the correct value to prime the hashing algorithm. The combination of integrity and authenticity assurance greatly improves the security profile of a system.

Architecture Description

As shown in Figure 1, there are two ROM's in this general architecture. The *Program ROM* contains all of the normal program instructions and the *Program Signature ROM* contains all of the signatures corresponding to pages in the *Program ROM*. Data throughput is improved by physically separating the *Program* and *Program Signature ROMs.* The *Prediction Module* calculates addresses that the authentication block should fetch based on addresses requested by the processor and cache misses. The *Program ROM* is dual ported to provide dedicated access of its contents to the *Prediction Module*. Once this address list is populated, the *Request Data and Signatures* block will send an address request to both ROM's. The *Program ROM* returns the cache page at the address requested and the *Program Signature ROM* returns the cache page's pre-computed signature.

The cache page is stored in the *Data Buffer* while the signature is stored in the *Signature Buffer*. A pre-agreed *Secure Hash Function*, using a pre-agreed IV, hashes the cache page residing in the *Data Buffer*. The result of this hash is compared to the pre-computed hash in the *Signature Buffer*. The *Signature Comparison* block in Figure 1 represents this operation.

A page leaves the right side of the comparison in Figure 1 and connects to a triangle. This same triangle receives a line from the *Data Buffer* on its left side. This is meant to represent the fact that the requested data does not cross through the gate (the triangle in this case), and into the cache, unless the *Signature Comparison* block opens the gate by activating the line leaving its right side. The line is activated when the pre-computed hash is identical to the calculated hash.

The *Cache Module* consists of three separate parts:

1. *Page Replacement Policy* – The algorithm responsible for evicting cache pages.
2. *Standard Cache* – The main cache holding groups of instructions for the processor to execute.
3. *Constant Cache* – The constants are placed in a separate cache because data inside of it is accessed randomly and frequently. This region should not be influenced by the page replacement policy for efficiency reasons.

The final architectural piece is the processor itself. It executes instructions and requests instructions from the cache. The cache is responsible for requesting data from the ROM that it does not yet have, on behalf of the processor.

Architectural Alternatives

There are several alternatives available that become tradeoffs (such as performance vs. resource). Some of these tradeoffs include:

1. Removing the *Prediction Module*
2. Using a single port ROM for *Program ROM*
3. Inserting the signatures directly into the *Program ROM*
4. Using a digital signature rather than a *Secure Hash Function*
5. Removing the *Constant Cache*

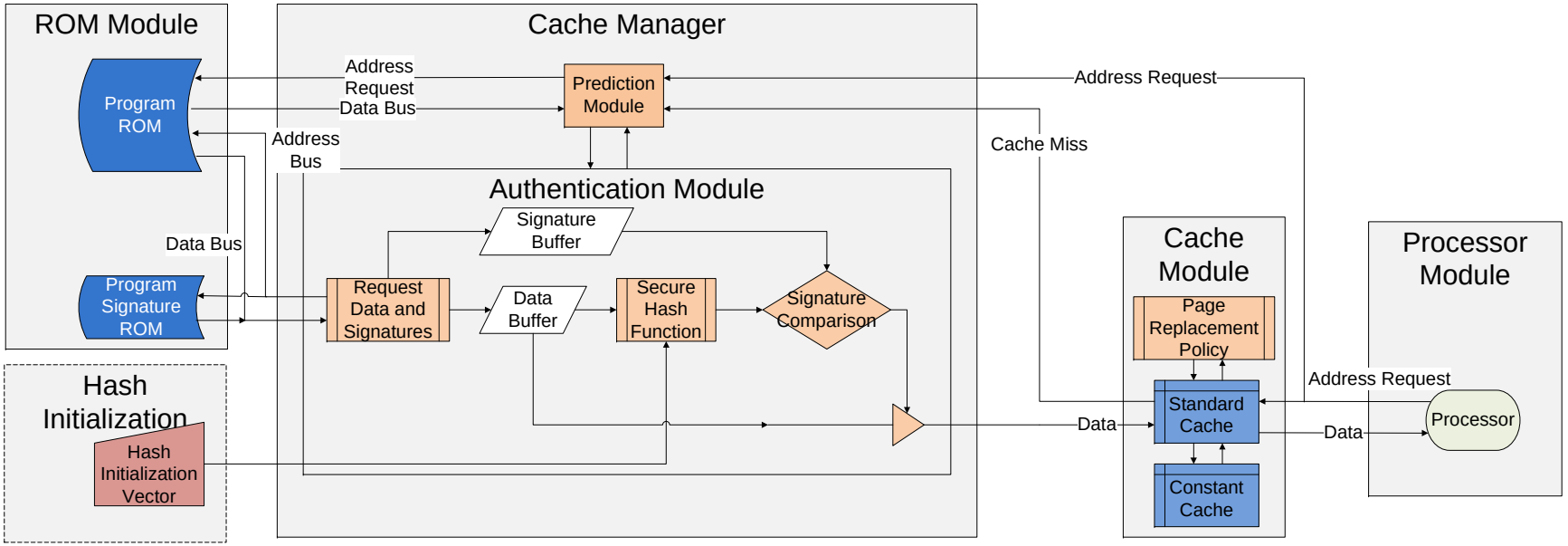

Figure 1: Authenticating Cache - predictive architecture

ELECTRONIC DISTRIBUTION:

1 J. A. MCCOY
31 Camino de Avila
Tijeras, NM 87059

1	MS 0114	R. T. WESTERVELT, 01931
1	MS 0161	D. J. JENKINS, 11500
1	MS 0451	S. M. BECKER, 02144
1	MS 0451	T. B. SMITH, 02144
1	MS 0453	B. L. REMUND, 2140
1	MS 0487	A. L. HILLHOUSE, 02142
1	MS 0899	Technical Library, 09536
1	MS 0359	D. Chavez, LDRD Office, 01911


Sandia National Laboratories