



**NAVAL
POSTGRADUATE
SCHOOL**

MONTEREY, CALIFORNIA

SP Summary (with Authority Mode)

by

Timothy E. Levin

18 September 2007

Approved for public release; distribution is unlimited.

Prepared for NSF and DARPA

This page left intentionally blank

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

Daniel T. Oliver
President

Leonard A. Ferrari
Executive Vice President and
Provost

This report was prepared for and funded by National Science Foundation and the Defense Advanced Research Projects Agency.

Reproduction of all or part of this report is authorized.

This report was prepared by:

Timothy E. Levin
Research Associate Professor

Reviewed by:

Released by:

Peter J. Denning
Department of Computer Science

Dan C. Boger
Interim Vice President and
Dean of Research

This page left intentionally blank

REPORT DOCUMENTATION PAGE			Form approved OMB No 0704-0188		
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.					
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 21 November 2007	3. REPORT TYPE AND DATES COVERED Research; 10/1/06 – 10/1/07		
4. TITLE AND SUBTITLE SP Summary (with Authority Mode)			5. FUNDING Grant number: CNS-0430566 and CNS-0430598		
6. AUTHOR(S) Timothy E. Levin					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Center for Information Systems Security Studies and Research (NPS CISR) 1411 Cunningham Rd., Monterey, CA 93943			8. PERFORMING ORGANIZATION REPORT NUMBER NPS-CS-08-007		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Science Foundation, 4201 Wilson Blvd. 1175 N. ArlingtonVA22230 DARPA, 3701 Fairfax Drive, Arlington, VA 22203			10. SPONSORING/MONITORING AGENCY REPORT NUMBER Not applicable		
11. SUPPLEMENTARY NOTES The views expressed in this report are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.					
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words.) This report provides summary of the interface and semantics for the processor extensions defined by the SP Processor.					
14. SUBJECT TERMS Operating systems: Hardware: Register-Transfer-Level Implementation: Control design; Security			15. NUMBER OF PAGES 17		
			16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU		

This page left intentionally blank



NAVAL POSTGRADUATE SCHOOL
CENTER FOR INFORMATION SYSTEMS SECURITY STUDIES AND RESEARCH



SecureCore

Trustworthy Commodity Computation and
Communication

| Technical Report: NPS-CS-08-007

SP Summary (with Authority Mode)

Timothy E. Levin

September 18, 2007

This page left intentionally blank

This material is based upon work supported by the National Science Foundation under Grant No. CNS-0430566 and CNS-0430598 with support from DARPA ATO. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or of DARPA ATO.

Author Affiliations

Naval Postgraduate School:

Timothy E. Levin
Center for Information Systems Security Studies and Research
Computer Science Department
Naval Postgraduate School
Monterey, California 93943

This page left intentionally blank

Abstract

This report provides summary of the interface and semantics for the processor extensions defined by the SP Processor [1][2] (where SP is the name of the design, which informally stands for “secret protected”).

SP Summary (with Authority Mode)

Timothy E. Levin

A. SP modes, modules and processing verification

- A processor *mode* is entered via a corresponding SP “begin” instruction
- SP is in a mode IFF SP is executing the corresponding type of *module*
- Module instructions checked via inline hashes and corresponding key:

Instruction	Module	Hash Key
BEGIN_A-CEM	A-TSM	DRK
BEGIN_U-CEM	U-TSM	DMK
BEGIN_CIC	I-TSM	DRK

- On return from an interrupt, InterruptHash of previous registers (uses DRK/DMK), and InterruptAddr (previous instruction) are checked; both values can be *saved* and *restored* by ring -2 to multiplex modes. Separate hash and addr values may be provided for each mode (A-TSM is not yet decided).

B. SP-resident master secrets – arbitrary values, 2-words¹ each

- **UserMasterKey** – UMK – read by UTSM; written only by lowest ring. Volatile storage
- **DeviceRootKey** – DRK – stored by “secure bios,” and locked until the next power cycle.² Non-volatile storage.
- **StorageRootHash**– SRH – read and written by ATSM. Non-volatile storage.
- **DeviceMasterKey** – DMK – stored by “secure bios,” and locked until the next power cycle.² Non-volatile storage.

C. SP transformation functions

- **Derive()** – 2-word to 2-word crypto-hash function available to ATSM
 - i. Based on DRK
- **CEM Load/Store()** – Available to ATSM/UTSM
 - i. Encrypt & hash one *word* on exit from processor cache– decrypt and check hash on load
 - ii. Based on DRK/DMK

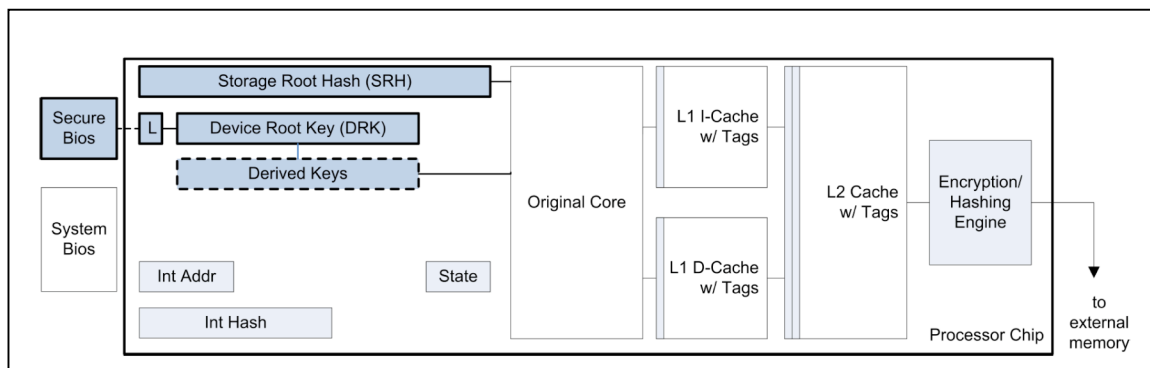


Figure 1. Authority Mode Features

¹ Word size depends on the architecture: e.g., 32 or 64 bits, and whether multiple load instructions are to be used.

² Restriction to lowest ring may also figure into DRK and DMK modifications.

Table 2. SP Instructions and Parameters

Instruction	Description
<u>Existing Startup SP Instructions (at secure bootup – secure BIOS only)</u>	
GRtoDMK R ₁ ,R ₂ , (DMK)	DMK = R ₁ R ₂ ; Sets Device Master Key register from GRs. (Only in Secure bootup BIOS – before any ring protection is established.)
<u>Existing SP Instructions</u>	
BEGIN_CEM	Enter CEM for next instruction. Sets CEM mode for next instruction.
END_CEM (only in CEM mode)	End CEM for next instruction. Clears CEM mode for next instruction
UMKtoGR (UMK), R ₁ ,R ₂ (only in CEM mode)	R ₁ R ₂ = UMK; Reads User Master Key register into GRs.
SECURE_STORE R ₁ , displ, R ₂ (only in CEM mode)	M[R ₁ + displ] = R ₂ ; Secure store from R ₂ to memory. Sets SecureData cache tag bits in on-chip caches.
SECURE_LOAD R ₁ , displ, R ₂ (only in CEM mode)	R ₂ = M[R ₁ + displ]; Secure load to R ₂ from memory. If hit in on-chip L1 Data cache or on-chip L2 cache, checks that SecureData tag bit is set for cache line, if not, evict and treat as a miss. If miss in on-chip caches, activate decryption and validation on fetching cache line from memory, raise exception if invalid.
<u>New Instructions to Enable Virtualization of SP (Ring -2 ONLY, non-CEM)</u>	
Save SPregs (Ring -2 only, non-CEM)	Copies SP.inthash, SP.retaddr and SP.status registers to secure space accessible ONLY to LPSK at Ring -2. Clears SP.inthash and SP.retaddr addresses, and sets SP.status appropriately. Done by LPSK only when LPSK switches between VMs.
Restore SPregs (Ring -2 only, non-CEM)	Restores SP.inthash, SP.retaddr and SP.status registers from secure space for this VM accessible ONLY to LPSK at Ring -2. Done by LPSK only when LPSK switches between VMs.

Table 3. Authority Mode Instructions and Parameters

Instruction	Operation	Description
<u>Authority Mode Instructions (new)</u>		
GR_TO_DAK R1,R2,(DAK)	DAK = R1 R2	Sets Device Attestation Key register from GRs.
DAK_LOCK (DAK_Lock)	DAK_Lock = 1	Sets DAK_Lock register to 1, disabling GR_TO_DAK instruction.
GR_TO_LSH R1,R2, (LSH)	LSH = R1 R2	Sets Local Storage Hash register from GRs.
LSH_TO_GR (LSH), R1,R2	R1 R2 = LSH	Reads the Local Storage Hash register into GRs.
DAK_DERIVE R1,R2,R3,(R4)	R3 R4 = HMAC _{DAK} (R1 R2)	Derives a key from the DAK. R1 R2 is the nonce. R3 R4 is the destination. (R4 is implied even register with R3.)
<u>Common Instructions (leveraged from SP)</u>		
BEGIN_CEM	CEMStatus = 01	Enter CEM for next instruction.
END_CEM	CEMStatus = 00	End CEM for next instruction.
CEM_STORE R1, displ, R2	M[R1 + displ] = R2	Secure store from GR to memory.
CEM_LOAD R1, displ, R2	R2 = M[R1 + displ]	Secure load to GR from memory.
<u>User Mode Instructions in SP (discarded)</u>		
GR_TO_DMK R1,R2, (DMK)	DMK = R1 R2	Sets Device Master Key register from GRs.
UMK_TO_GR (UMK), R1,R2	R1 R2 = UMK	Reads User Master Key register into GRs.

Note that Table 3 uses obsolete names “Local Storage Hash” (SRH) and “DAK, which have been changed to “Storage Root Hash” (SRH), and DRK, respectively.

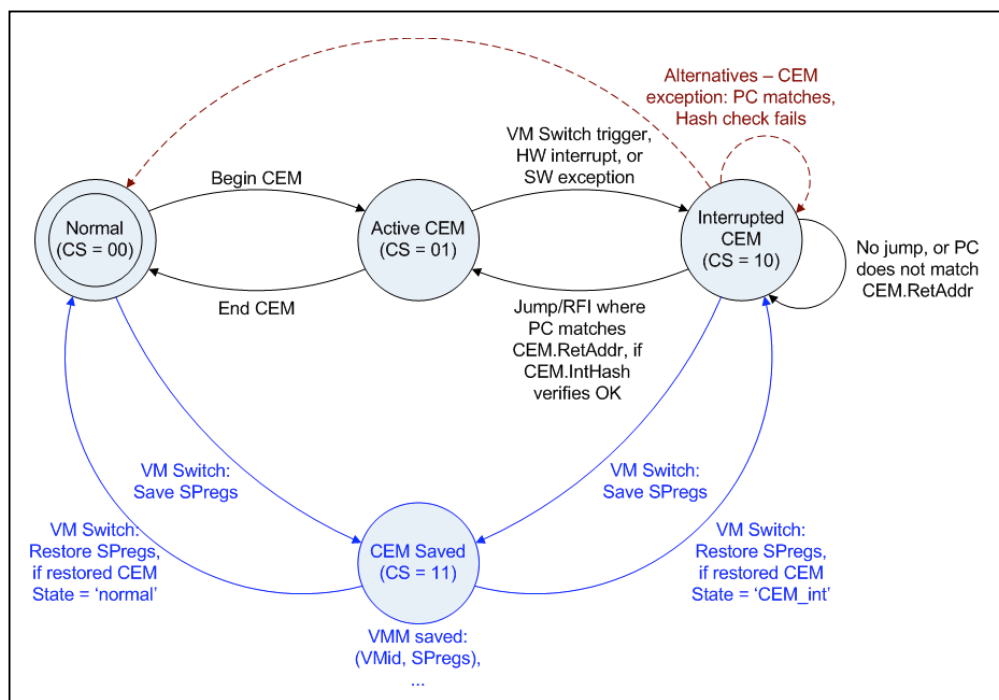


Figure 1. SP State Diagram (User Mode; Auth Mode not yet clear)

Table 4. SP Internal Transformations (User and Authority Mode)

Instruction	Prerequisite State	SP actions	Post State	Required Post-processing	Description
Request for execution context change ³	Active_CEM	- Encrypt registers in place - Store hash ⁴ of concatenated, encrypted registers to CEM.IntHash - Store PC to CEM.RetAddr - Load and process interrupt vector	Interrupted_CEM	Software indicated by interrupt vector saves PC and GP registers	Preserves CEM state on context switch to non-TSM code
Request for execution context change ⁵	- Interrupted_CEM - Previous PC and GP registers loaded by software	- If PC matches CEM.RetAddr ⁶ then (-- if hash of GP registers matches CEM.IntHash then (--- Decrypt registers in place; --- process PC))	Active_CEM	None	Restores CEM state on return from non-TSM code

³ HW interrupt or software exception

⁴ IV for the register encryption will likely be stored with the hash.

⁵ Any HW Jump or return from interrupt

⁶ SP designers intend to introduce a feature to prevent accidental return to the PC address from a different address space, which could be handled in a few different ways, depending on OS support. If hash check fails, SP will raise an exception.

D. References

- [1] R. B. Lee, P. C. S. Kwan, J. P. McGregor, J. Dwoskin, and Z. Wang, "Architecture for protecting critical secrets in microprocessors," in *ISCA '05: Proceedings of the 32nd annual international symposium on Computer Architecture*, (Washington, DC, USA), pp. 2–13, IEEE Computer Society, 2005.
- [2] J. Dwoskin and R. B. Lee, "Hardware-rooted Trust for Secure Key Management and Transient Trust," *Proc. ACM Conf. Computer Commun. Security*, pp. 389-400, Oct 2007.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 1
Ft. Belvoir, VA
2. Dudley Knox Library 1
Naval Postgraduate School
Monterey, CA
3. Lee Badger 1
DARPA
Arlington, VA 22203
4. Terry V. Benzel 1
Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
5. Ganesha Bhaskara 1
Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
6. Paul C. Clark 2
Naval Postgraduate School
Monterey, CA
7. Cynthia E. Irvine 2
Naval Postgraduate School
Monterey, CA
8. Timothy E. Levin 2
Naval Postgraduate School
Monterey, CA
9. Karl Levitt 1
National Science Foundation
4201 Wilson Blvd.
Arlington, VA 22230
10. Thuy D. Nguyen 2
Naval Postgraduate School
Monterey, CA

This page left intentionally blank