David Pointcheval (Ed.)

LNCS 3860

Topics in Cryptology – CT-RSA 2006

The Cryptographers' Track at the RSA Conference 2006
San Jose, CA, USA, February 2006
Proceedings

RSA™ Conference 2006

Springer

David Pointcheval (Ed.)

Topics in Cryptology – CT-RSA 2006

The Cryptographers' Track at the RSA Conference 2006
San Jose, CA, USA, February 13-17, 2006
Proceedings

Springer

Volume Editor

David Pointcheval
CNRS
ENS/DI
45, rue d'Ulm, 75005 Paris, France
E-mail: David.Pointcheval@ens.fr

Library of Congress Control Number: 2005937532

CR Subject Classification (1998): E.3, G.2.1, D.4.6, K.6.5, K.4.4, F.2.1-2, C.2, J.1

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-540-31033-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-31033-4 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11605805 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3860

Lecture Notes in Computer Science

For information about Vols. 1–3744

please contact your bookseller or Springer

Vol. 3860: D. Pointcheval (Ed.), Topics in Cryptology – CT-RSA 2006. XI, 365 pages. 2006.

Vol. 3850: R. Freund, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 371 pages. 2006.

Vol. 3838: A. Middeldorp, V. van Oostrom, F. van Raamsdonk, R. de Vrijer (Eds.), Processes, Terms and Cycles: Steps on the Road to Infinity. XVIII, 639 pages. 2005.

Vol. 3837: K. Cho, P. Jacquet (Eds.), Technologies for Advanced Heterogeneous Networks. IX, 307 pages. 2005.

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005. (Sublibrary LNAI).

Vol. 3833: K.-J. Li, C. Vangenot (Eds.), Web and Wireless Geographical Information Systems. XI, 309 pages. 2005.

Vol. 3829: P. Pettersson, W. Yi (Eds.), Formal Modeling and Analysis of Timed Systems. IX, 305 pages. 2005.

Vol. 3828: X. Deng, Y. Ye (Eds.), Internet and Network Economics. XVII, 1106 pages. 2005.

Vol. 3827: X. Deng, D. Du (Eds.), Algorithms and Computation. XX, 1190 pages. 2005.

Vol. 3826: B. Benatallah, F. Casati, P. Traverso (Eds.), Service-Oriented Computing - ICSOC 2005. XVIII, 597 pages. 2005.

Vol. 3824: L.T. Yang, M. Amamiya, Z. Liu, M. Guo, F.J. Rammig (Eds.), Embedded and Ubiquitous Computing – EUC 2005. XXIII, 1204 pages. 2005.

Vol. 3823: T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai, L.T. Yang (Eds.), Embedded and Ubiquitous Computing – EUC 2005 Workshops. XXXII, 1317 pages. 2005.

Vol. 3822: D. Feng, D. Lin, M. Yung (Eds.), Information Security and Cryptology. XII, 420 pages. 2005.

Vol. 3821: R. Ramanujam, S. Sen (Eds.), FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science. XIV, 566 pages. 2005.

Vol. 3820: L.T. Yang, X. Zhou, W. Zhao, Z. Wu, Y. Zhu, M. Lin (Eds.), Embedded Software and Systems. XXVIII, 779 pages. 2005.

Vol. 3819: P. Van Hentenryck (Ed.), Practical Aspects of Declarative Languages. X, 231 pages. 2006.

Vol. 3818: S. Grumbach, L. Sui, V. Vianu (Eds.), Advances in Computer Science – ASIAN 2005. XIII, 294 pages. 2005.

Vol. 3815: E.A. Fox, E.J. Neuhold, P. Premsmit, V. Wuwongse (Eds.), Digital Libraries: Implementing Strategies and Sharing Experiences. XVII, 529 pages. 2005.

Vol. 3814: M. Maybury, O. Stock, W. Wahlster (Eds.), Intelligent Technologies for Interactive Entertainment. XV, 342 pages. 2005. (Sublibrary LNAI).

Vol. 3813: R. Molva, G. Tsudik, D. Westhoff (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. VIII, 219 pages. 2005.

Vol. 3810: Y.G. Desmedt, H. Wang, Y. Mu, Y. Li (Eds.), Cryptology and Network Security. XI, 349 pages. 2005.

Vol. 3809: S. Zhang, R. Jarvis (Eds.), AI 2005: Advances in Artificial Intelligence. XXVII, 1344 pages. 2005. (Sublibrary LNAI).

Vol. 3808: C. Bento, A. Cardoso, G. Dias (Eds.), Progress in Artificial Intelligence. XVIII, 704 pages. 2005. (Sublibrary LNAI).

Vol. 3807: M. Dean, Y. Guo, W. Jun, R. Kaschek, S. Krishnaswamy, Z. Pan, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005 Workshops. XV, 275 pages. 2005.

Vol. 3806: A.H. H. Ngu, M. Kitsuregawa, E.J. Neuhold, J.-Y. Chung, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005. XXI, 771 pages. 2005.

Vol. 3805: G. Subsol (Ed.), Virtual Storytelling. XII, 289 pages. 2005.

Vol. 3804: G. Bebis, R. Boyle, D. Koracin, B. Parvin (Eds.), Advances in Visual Computing. XX, 755 pages. 2005.

Vol. 3803: S. Jajodia, C. Mazumdar (Eds.), Information Systems Security. XI, 342 pages. 2005.

Vol. 3802: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part II. XLII, 1166 pages. 2005. (Sublibrary LNAI).

Vol. 3801: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part I. XLI, 1122 pages. 2005. (Sublibrary LNAI).

Vol. 3799: M. A. Rodríguez, I.F. Cruz, S. Levashkin, M.J. Egenhofer (Eds.), GeoSpatial Semantics. X, 259 pages. 2005.

Vol. 3798: A. Dearle, S. Eisenbach (Eds.), Component Deployment. X, 197 pages. 2005.

Vol. 3797: S. Maitra, C. E. V. Madhavan, R. Venkatesan (Eds.), Progress in Cryptology - INDOCRYPT 2005. XIV, 417 pages. 2005.

Vol. 3796: N.P. Smart (Ed.), Cryptography and Coding. XI, 461 pages. 2005.

Vol. 3795: H. Zhuge, G.C. Fox (Eds.), Grid and Cooperative Computing - GCC 2005. XXI, 1203 pages. 2005.

Vol. 3794: X. Jia, J. Wu, Y. He (Eds.), Mobile Ad-hoc and Sensor Networks. XX, 1136 pages. 2005.

Vol. 3793: T. Conte, N. Navarro, W.-m.W. Hwu, M. Valero, T. Ungerer (Eds.), High Performance Embedded Architectures and Compilers. XIII, 317 pages. 2005.

Vol. 3792: I. Richardson, P. Abrahamsson, R. Messnarz (Eds.), Software Process Improvement. VIII, 215 pages. 2005.

Vol. 3791: A. Adi, S. Stoutenburg, S. Tabet (Eds.), Rules and Rule Markup Languages for the Semantic Web. X, 225 pages. 2005.

Vol. 3790: G. Alonso (Ed.), Middleware 2005. XIII, 443 pages. 2005.

Vol. 3789: A. Gelbukh, Á. de Albornoz, H. Terashima-Marín (Eds.), MICAI 2005: Advances in Artificial Intelligence. XXVI, 1198 pages. 2005. (Sublibrary LNAI).

Vol. 3788: B. Roy (Ed.), Advances in Cryptology - ASIACRYPT 2005. XIV, 703 pages. 2005.

Vol. 3785: K.-K. Lau, R. Banach (Eds.), Formal Methods and Software Engineering. XIV, 496 pages. 2005.

Vol. 3784: J. Tao, T. Tan, R.W. Picard (Eds.), Affective Computing and Intelligent Interaction. XIX, 1008 pages. 2005.

Vol. 3783: S. Qing, W. Mao, J. Lopez, G. Wang (Eds.), Information and Communications Security. XIV, 492 pages. 2005.

Vol. 3781: S.Z. Li, Z. Sun, T. Tan, S. Pankanti, G. Chollet, D. Zhang (Eds.), Advances in Biometric Person Authentication. XI, 250 pages. 2005.

Vol. 3780: K. Yi (Ed.), Programming Languages and Systems. XI, 435 pages. 2005.

Vol. 3779: H. Jin, D. Reed, W. Jiang (Eds.), Network and Parallel Computing. XV, 513 pages. 2005.

Vol. 3778: C. Atkinson, C. Bunse, H.-G. Gross, C. Peper (Eds.), Component-Based Software Development for Embedded Systems. VIII, 345 pages. 2005.

Vol. 3777: O.B. Lupanov, O.M. Kasim-Zade, A.V. Chaskin, K. Steinhöfel (Eds.), Stochastic Algorithms: Foundations and Applications. VIII, 239 pages. 2005.

Vol. 3776: S.K. Pal, S. Bandyopadhyay, S. Biswas (Eds.), Pattern Recognition and Machine Intelligence. XXIV, 808 pages. 2005.

Vol. 3775: J. Schönwälder, J. Serrat (Eds.), Ambient Networks. XIII, 281 pages. 2005.

Vol. 3774: G. Bierman, C. Koch (Eds.), Database Programming Languages. X, 295 pages. 2005.

Vol. 3773: A. Sanfeliu, M.L. Cortés (Eds.), Progress in Pattern Recognition, Image Analysis and Applications. XX, 1094 pages. 2005.

Vol. 3772: M. Consens, G. Navarro (Eds.), String Processing and Information Retrieval. XIV, 406 pages. 2005.

Vol. 3771: J.M.T. Romijn, G.P. Smith, J. van de Pol (Eds.), Integrated Formal Methods. XI, 407 pages. 2005.

Vol. 3770: J. Akoka, S.W. Liddle, I.-Y. Song, M. Bertolotto, I. Comyn-Wattiau, W.-J. van den Heuvel, M. Kolp, J. Trujillo, C. Kop, H.C. Mayr (Eds.), Perspectives in Conceptual Modeling. XXII, 476 pages. 2005.

Vol. 3769: D.A. Bader, M. Parashar, V. Sridhar, V.K. Prasanna (Eds.), High Performance Computing – HiPC 2005. XXVIII, 550 pages. 2005.

Vol. 3768: Y.-S. Ho, H.J. Kim (Eds.), Advances in Multimedia Information Processing - PCM 2005, Part II. XXVIII, 1088 pages. 2005.

Vol. 3767: Y.-S. Ho, H.J. Kim (Eds.), Advances in Multimedia Information Processing - PCM 2005, Part I. XXVIII, 1022 pages. 2005.

Vol. 3766: N. Sebe, M.S. Lew, T.S. Huang (Eds.), Computer Vision in Human-Computer Interaction. X, 231 pages. 2005.

Vol. 3765: Y. Liu, T. Jiang, C. Zhang (Eds.), Computer Vision for Biomedical Image Applications. X, 563 pages. 2005.

Vol. 3764: S. Tixeuil, T. Herman (Eds.), Self-Stabilizing Systems. VIII, 229 pages. 2005.

Vol. 3762: R. Meersman, Z. Tari, P. Herrero (Eds.), On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops. XXXI, 1228 pages. 2005.

Vol. 3761: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Part II. XXVII, 653 pages. 2005.

Vol. 3760: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Part I. XXVII, 921 pages. 2005.

Vol. 3759: G. Chen, Y. Pan, M. Guo, J. Lu (Eds.), Parallel and Distributed Processing and Applications - ISPA 2005 Workshops. XIII, 669 pages. 2005.

Vol. 3758: Y. Pan, D.-x. Chen, M. Guo, J. Cao, J.J. Dongarra (Eds.), Parallel and Distributed Processing and Applications. XXIII, 1162 pages. 2005.

Vol. 3757: A. Rangarajan, B. Vemuri, A.L. Yuille (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition. XII, 666 pages. 2005.

Vol. 3756: J. Cao, W. Nejdl, M. Xu (Eds.), Advanced Parallel Processing Technologies. XIV, 526 pages. 2005.

Vol. 3754: J. Dalmau Royo, G. Hasegawa (Eds.), Management of Multimedia Networks and Services. XII, 384 pages. 2005.

Vol. 3753: O.F. Olsen, L.M.J. Florack, A. Kuijper (Eds.), Deep Structure, Singularities, and Computer Vision. X, 259 pages. 2005.

Vol. 3752: N. Paragios, O. Faugeras, T. Chan, C. Schnörr (Eds.), Variational, Geometric, and Level Set Methods in Computer Vision. XI, 369 pages. 2005.

Vol. 3751: T. Magedanz, E.R.M. Madeira, P. Dini (Eds.), Operations and Management in IP-Based Networks. X, 213 pages. 2005.

Vol. 3750: J.S. Duncan, G. Gerig (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005, Part II. XL, 1018 pages. 2005.

Vol. 3749: J.S. Duncan, G. Gerig (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005, Part I. XXXIX, 942 pages. 2005.

Vol. 3748: A. Hartman, D. Kreische (Eds.), Model Driven Architecture – Foundations and Applications. IX, 349 pages. 2005.

Vol. 3747: C.A. Maziero, J.G. Silva, A.M.S. Andrade, F.M.d. Assis Silva (Eds.), Dependable Computing. XV, 267 pages. 2005.

Vol. 3746: P. Bozanis, E.N. Houstis (Eds.), Advances in Informatics. XIX, 879 pages. 2005.

Vol. 3745: J.L. Oliveira, V. Maojo, F. Martín-Sánchez, A.S. Pereira (Eds.), Biological and Medical Data Analysis. XII, 422 pages. 2005. (Sublibrary LNBI).

¥444.00元

Preface

The RSA® Conference, with over 15,000 attendees, as well as over 225 sponsors and exhibitors, is the largest computer security event of the year. The Cryptographers' Track is one of the many parallel tracks. These proceedings contain the papers presented during the sixth edition. The tradition indeed started in 2001, and is by now well established: the Cryptographers' Track at the RSA Conference is among the major events in cryptography.

There were 72 submitted contributions, of which 22 were selected for presentation. They cover all aspects of cryptography (symmetric and asymmetric cryptography, constructions and attacks, new trends). In addition, the program includes two invited talks, by Xiaoyun Wang on "Cryptanalysis of Hash functions and Potential Dangers," and Philip MacKenzie on "Passwords Will Not Die: How Cryptography Can Help Deal with Them."

All the submissions were reviewed by at least three members of the Program Committee. I am very grateful to the 24 members for their hard and conscientious work. Many thanks to the 89 external reviewers:

Masayuki Abe
Kazumaro Aoki
Giuseppe Ateniese
Roberto Avanzi
Zuzana Beerliová
Olivier Billet
Alex Biryukov
Ian Blake
Colin Boyd
Eric Brier
Aniello Castiglione
Juyoung Cha
Aldar Chan
Liqun Chen
Kookrae Cho
Scott Contini
Paolo D'Arco
Jintai Ding
Christophe Doche
Orr Dunkelman
Matthias Fitzi
Pierre-Alain Fouque
Jacques J.A. Fournier
Kouichi Fujisaki

Eiichiro Fujisaki
Jun Furukawa
David Galindo
Shai Halevi
Helena Handschuh
Chris Heneghan
Thomas Holenstein
Fumitaka Hoshino
Yong Ho Hwang
Toshiyuki Isshiki
Ellen Jochemsz
Antoine Joux
Ari Juels
Charanjit Jutla
Aggelos Kiayias
Hiroaki Kikuchi
Tetsutarou Kobayashi
Tadayoshi Kohno
Hugo Krawczyk
Sandeep Kumar
Tanja Lange
Jung Wook Lee
Barbara Masucci
Alexander May

Miodrag Mihaljevic
Kazuhiko Minematsu
Fabian Monrose
Paul Montague
Steve Myers
David Naccache
Antonio Nicolosi
Satoshi Obana
Satomi Okazaki
Katsuyuki Okeya
Francis Olivier
Roger Oyono
Dan Page
Jung Hyung Park
Kun Peng
Krzysztof Pietrzak
Dominik Raub
Yasuyuki Sakai
Kouichi Sakurai
Werner Schindler
Jae Woo Seo
Jong Hoon Shin
Igor Shparlinski
Ron Steinfeld

Mike Szydlo
Yael Tauman Kalai
Isamu Teranishi
Toshio Tokita
Michael Tunstall
Frederik Vercauteren
Karine Villegas
Shabsi Walfish
Huaxiong Wang
Xiaofeng Wang
Bogdan Warinschi
Benne de Weger
Christopher Wolf
Alex Yampolskiy
Yeon Hyeong Yang
Yiqun Lisa Yin
Jeong Il Yoon

Note that these proceedings contain the revised versions of the selected papers. Since the revisions were not checked again before publication, the authors (and not the committee) bear full responsibility of the contents of their papers.

I also would like to thank Jacques Beigbeder for maintaining the submission and webreview servers, and Duong Hieu Phan for the fast set up of the review phase. The submission software was written by Chanathip Namprempre, and the webreview system by Wim Moreau and Joris Claessens. Many thanks to Burt Kaliski for interfacing with the RSA conference organizers, and to Alfred Hofmann at Springer for the production of this volume.

Finally, I wish to thank all the authors who submitted papers, and the authors of accepted papers for sending their final versions on time.

November 2005

David Pointcheval
Program Chair
CT-RSA 2006

Organization

RSA Conference 2006 was organized by RSA Security Inc. and its partner organizations around the world. The Cryptogaphers' Track at RSA Conference 2006 was organized by RSA Laboratories (http://www.rsasecurity.com).

Program Chair

David Pointcheval	CNRS/ENS, France

Program Committee

Eli Biham	Technion, Israel
Xavier Boyen	Voltage, USA
Benoît Chevallier-Mames	Gemplus, France
Anand Desai	NTT MCL, USA
Yvo Desmedt	University College London, UK
Yevgeniy Dodis	New York Univ., USA
Steven Galbraith	Royal Holloway University of London, UK
Rosario Gennaro	IBM T.J. Watson Research Center, USA
Henri Gilbert	France Telecom R&D, France
Martin Hirt	ETH Zurich, Switzerland
Nick Howgrave-Graham	NTRU Cryptosystems, USA
Markus Jakobsson	Indiana Univ., USA
Jonathan Katz	Univ. of Maryland, USA
Kwangjo Kim	ICU, Korea
Pil Joong Lee	POSTECH, Korea
Arjen Lenstra	Lucent Technologies, USA & TU Eindhoven, The Netherlands
Javier Lopez	Univ. of Malaga, Spain
Tatsuaki Okamoto	NTT, Japan
Josef Pieprzyk	Macquarie Univ., Australia
Guillaume Poupard	DCSSI Crypto Lab, France
Bart Preneel	K.U. Leuven, Belgium
Kazue Sako	NEC, Japan
Ivan Visconti	Univ. di Salerno, Italy
Moti Yung	RSA Labs & Columbia Univ., USA

Table of Contents

Attacks on AES

Cache Attacks and Countermeasures: The Case of AES
Dag Arne Osvik, Adi Shamir, Eran Tromer 1

Related-Key Impossible Differential Attacks on 8-Round AES-192
Eli Biham, Orr Dunkelman, Nathan Keller 21

Identification

Session Corruption Attack and Improvements on Encryption Based MT-Authenticators
Xiaojian Tian, Duncan S. Wong 34

Fair Identification
Omkant Pandey, Julien Cathalo, Jean-Jacques Quisquater 52

Algebra

Efficient Doubling on Genus 3 Curves over Binary Fields
Xinxin Fan, Thomas Wollinger, Yumin Wang 64

Another Look at Small RSA Exponents
M. Jason Hinek 82

Integrity

Collision-Resistant Usage of MD5 and SHA-1 Via Message Preprocessing
Michael Szydlo, Yiqun Lisa Yin 99

RFID-Tags for Anti-counterfeiting
Pim Tuyls, Lejla Batina 115

Public Key Encryption

A "Medium-Field" Multivariate Public-Key Encryption Scheme
Lih-Chung Wang, Bo-Yin Yang, Yuh-Hua Hu, Feipei Lai 132

A New Security Proof for Damgård's ElGamal
Kristian Gjøsteen 150

Signatures

Stand-Alone and Setup-Free Verifiably Committed Signatures
Huafei Zhu, Feng Bao 159

Toward the Fair Anonymous Signatures: Deniable Ring Signatures
Yuichi Komano, Kazuo Ohta, Atsushi Shimbo, Shinichi Kawamura 174

Side-Channel Attacks

Practical Second-Order DPA Attacks for Masked Smart Card Implementations of Block Ciphers
Elisabeth Oswald, Stefan Mangard, Christoph Herbst, Stefan Tillich 192

Higher Order Masking of the AES
Kai Schramm, Christof Paar 208

CCA Encryption

Chosen Ciphertext Secure Public Key Threshold Encryption Without Random Oracles
Dan Boneh, Xavier Boyen, Shai Halevi 226

How to Construct Multicast Cryptosystems Provably Secure Against Adaptive Chosen Ciphertext Attack
Yitao Duan, John Canny 244

Message Authentication

On the (Im)possibility of Blind Message Authentication Codes
Michel Abdalla, Chanathip Namprempre, Gregory Neven 262

An Optimal Non-interactive Message Authentication Protocol
Sylvain Pasini, Serge Vaudenay 280

Block Ciphers

A New Criterion for Nonlinearity of Block Ciphers
Orr Dunkelman, Nathan Keller 295

Block Ciphers Sensitive to Gröbner Basis Attacks
Johannes Buchmann, Andrei Pyshkin, Ralf-Philipp Weinmann 313

Multi-party Computation

Universally Composable Oblivious Transfer in the Multi-party Setting
Marc Fischlin 332

A Round and Communication Efficient Secure Ranking Protocol
Shaoquan Jiang, Guang Gong 350

Author Index 365

Cache Attacks and Countermeasures: The Case of AES

Dag Arne Osvik[1], Adi Shamir[2], and Eran Tromer[2]

[1] dag.arne@osvik.no
[2] Department of Computer Science and Applied Mathematics,
Weizmann Institute of Science, Rehovot 76100, Israel
{adi.shamir, eran.tromer}@weizmann.ac.il

Abstract. We describe several software side-channel attacks based on inter-process leakage through the state of the CPU's memory cache. This leakage reveals memory access patterns, which can be used for cryptanalysis of cryptographic primitives that employ data-dependent table lookups. The attacks allow an unprivileged process to attack other processes running in parallel on the same processor, despite partitioning methods such as memory protection, sandboxing and virtualization. Some of our methods require only the ability to trigger services that perform encryption or MAC using the unknown key, such as encrypted disk partitions or secure network links. Moreover, we demonstrate an extremely strong type of attack, which requires knowledge of neither the specific plaintexts nor ciphertexts, and works by merely monitoring the effect of the cryptographic process on the cache. We discuss in detail several such attacks on AES, and experimentally demonstrate their applicability to real systems, such as OpenSSL and Linux's dm-crypt encrypted partitions (in the latter case, the full key can be recovered after just 800 writes to the partition, taking 65 milliseconds). Finally, we describe several countermeasures for mitigating such attacks.

Keywords: side-channel attack, cache, memory access, cryptanalysis, AES.

1 Introduction

1.1 Overview

Many computer systems concurrently execute programs with different privileges, employing various partitioning methods to facilitate the desired access control semantics. These methods include kernel vs. userspace separation, process memory protection, filesystem permissions and chroot, and various approaches to virtual machines and sandboxes. All of these rely on a model of the underlying machine to obtain the desired access control semantics. However, this model is often idealized and does not reflect many intricacies of actual implementation.

In this paper we show how a low-level implementation detail of modern CPUs, namely the structure of memory caches, causes subtle indirect interaction between processes running on the same processor. This leads to cross-process information leakage. In essence, the cache forms a shared resource which all processes

D. Pointcheval (Ed.): CT-RSA 2006, LNCS 3860, pp. 1–20, 2006.

compete for, and it thus affects and is affected by every process. While the *data* stored in the cache is protected by virtual memory mechanisms, the *metadata* about the contents of the cache, and hence the memory access patterns of processes using that cache, is not fully protected.

We describe several methods an attacker can use to learn about the memory access patterns of another process. These are classified into methods that affect the state of the cache and then measure the effect on the running time of the encryption, and methods that investigate the state of the cache after or during encryption. The latter are found to be particularly effective and noise-resistant.

We demonstrate the cryptanalytic applicability of these methods to the Advanced Encryption Standard (AES, [11]) by showing a known-plaintext (or known-ciphertext) attack that performs efficient full key extraction. For example, an implementation of one variant of the attack performs full AES key extraction from the `dm-crypt` system of Linux using only 800 accesses to an encrypted file, 65ms of measurements and 3 seconds of analysis; attacking simpler systems, such as "black-box" OpenSSL library calls, is even faster at 13ms and 300 encryptions.

One variant of our attack has the unusual property of performing key extraction *without knowledge of either the plaintext or the ciphertext.* This is an unusually strong form of attack in which an unprivileged process can, just by accessing its own memory space, obtain bits from a secret AES key used by another process, without any (explicit) communication between the two. This too is demonstrated experimentally.

Implementing AES in a way that is impervious to this attack, let alone developing an efficient generic countermeasure, appears non-trivial; in Section 5, various countermeasures are described and analyzed.

Many details and variants have been omitted due to space constraints; see `http://www.wisdom.weizmann.ac.il/~tromer/cache` for an extended version.

1.2 Related Works

The possibility of cross-process leakage via cache state has been mentioned in several previous works. It was considered in 1992 by Hu [7] in the context of intentional transmission via covert channels. In 1998, Kelsey et al. [8] mentioned the prospect of "attacks based on cache hit ratio in large S-box ciphers". In 2002, Page [9] described theoretical attacks using cache misses, but assumed the ability to identify cache misses with very high temporal resolution; its applicability in realistic scenarios is unclear. In 2003, Tsunoo et al. [15] described attacks using timing effects due to collisions in the memory lookups *inside* the cipher, as opposed to the cipher-attacker collisions we investigate.

Concurrently with but independently of our work, Bernstein [2] describes attacks on AES that exploit timing variability due to cache effects; his attack can be seen as a variant of our Evict+Time measurement method (see Section 3.4). The main difference is that [2] does not use an explicit model of the cache and active manipulation, but rather relies only on the existence of some consistent statistical timing pattern due to various uncontrolled memory access effects. The resulting attack is simpler and more portable, but have several shortcomings.

First, it requires reference measurements of encryption under *known* key in an identical configuration, and these are often not readily available (e.g., a user may be able to write data to an encrypted filesystem, but creating a reference filesystem with a known key is a privileged operation). Second, the attack of [2] relies on timing the encryption and thus, similarly to our Evict+Time method, seems impractical on many real systems due to excessively low signal-to-noise ratio; our alternative methods (Sections 3.5 and 4) address this. Third, even when the attack of [2] works, it requires a much higher number of analyzed encryptions.[1]

Also concurrently with but independently of our work, Percival [14] describes a cache-based attack on RSA for processors with simultaneous multithreading. The measurement method is similar to one variant of our asynchronous attack (Section 4), but the cryptanalytic aspect is very different since the algorithms and time scales involved in RSA encryption are very different from those of AES. Both [2] and [14] contain discussions of countermeasures against the respective attacks, and some of these are also relevant to our attacks (see Section 5).

Koeune and Quisquater [6] described a timing attack on a "bad implementation" of AES which uses its algebraic description in a "careless way" (namely, using a conditional branch in the MixColumn operation). That attack is not applicable to common software implementations, but should be taken into account in regard to certain countermeasures against our attack (see Section 5.2).

Leakage of memory access information has also been considered in other contexts, yielding theoretical [5] and practical [16][17] mitigation methods; these are discussed in Section 5.3.

2 Preliminaries

2.1 Memory and Cache Structure

Modern processors use one or more levels of *set-associative memory cache*. Such a cache consists of storage cells called *cache lines*, each consisting of B bytes. The cache is organized into S *cache sets*, each containing W cache lines[2], so overall the cache contains $S \cdot W \cdot B$ bytes. The mapping of memory addresses into the cache is limited as follows. First, the cache holds copies of aligned blocks of B bytes in main memory, which we will term *memory blocks*; when a cache miss occurs, a full memory block is copied into one of the cache lines. Second, each memory block may be cached only in a specific cache set; specifically, the memory block starting at address a can be cached only in the W cache lines belonging to cache set $\lfloor a/B \rfloor \bmod S$. See Figure 1(a). Thus, the memory blocks are partitioned into S classes, where the blocks in each class contend for the cache lines in a single cache set.

[1] In our experiments the attack code of [2] failed to get a signal from `dm-crypt` even after a 10 hours run, whereas in an identical setup our Prime+Probe performed full key recovery using 65ms of measurements.

[2] In common terminology, W is called the *associativity* and the cache is called *W-way associative*.

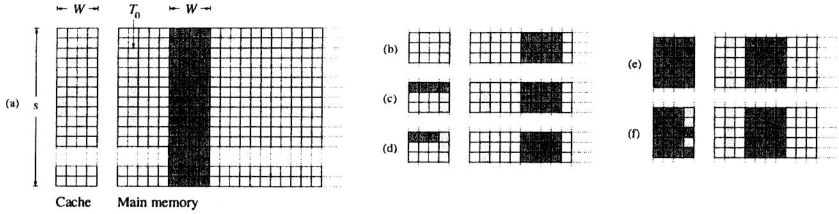

Fig. 1. (a) Schematic of a set-associative cache. The light gray blocks represent a cached AES lookup table. The dark gray blocks represent the attacker's memory. States (b)-(d) depict Evict+Time and (e)-(f) depict Prime+Probe (see Section 3).

2.2 Memory Access in AES Implementations

This paper focuses on AES (see Section 6.1 for a discussion of other ciphers). Performance-oriented implementations on 32-bit (or higher) processors typically use the following formulation, as prescribed in the Rijndael AES submission [4].[3]

Several lookup tables are precomputed once by the programmer or during system initialization. There are 8 such tables, $T_0, \ldots, T_3$ and $T_0^{(10)}, \ldots, T_3^{(10)}$, each containing 256 4-byte words. The contents of the tables, defined in [4], are inconsequential for most of our attacks.

During key setup, a given 16-byte secret key $\boldsymbol{k} = (k_0, \ldots, k_{15})$ is expanded into 10 round keys[4], $\boldsymbol{K}^{(r)}$ for $r = 1, \ldots, 10$. Each round key is divided into 4 words of 4 bytes each: $\boldsymbol{K}^{(r)} = (K_0^{(r)}, K_1^{(r)}, K_2^{(r)}, K_3^{(r)})$. The 0-th round key is just the raw key: $K_j^{(0)} = (k_{4j}, k_{4j+1}, k_{4j+2}, k_{4j+3})$ for $j = 0, 1, 2, 3$. The details of the rest of the expansion are mostly inconsequential.

Given a 16-byte plaintext $\boldsymbol{p} = (p_0, \ldots, p_{15})$, encryption proceeds by computing a 16-byte intermediate state $\boldsymbol{x}^{(r)} = (x_0^{(r)}, \ldots, x_{15}^{(r)})$ at each round r. The initial state $\boldsymbol{x}^{(0)}$ is computed by $x_i^{(0)} = p_i \oplus k_i$ $(i = 0, \ldots, 15)$. Then, the first 9 rounds are computed by updating the intermediate state as follows, for $r = 0, \ldots, 8$:

$$\begin{aligned}
(x_0^{(r+1)}, x_1^{(r+1)}, x_2^{(r+1)}, x_3^{(r+1)}) &\leftarrow T_0[x_0^{(r)}] \oplus T_1[x_5^{(r)}] \oplus T_2[x_{10}^{(r)}] \oplus T_3[x_{15}^{(r)}] \oplus K_0^{(r+1)} \\
(x_4^{(r+1)}, x_5^{(r+1)}, x_6^{(r+1)}, x_7^{(r+1)}) &\leftarrow T_0[x_4^{(r)}] \oplus T_1[x_9^{(r)}] \oplus T_2[x_{14}^{(r)}] \oplus T_3[x_3^{(r)}] \oplus K_1^{(r+1)} \\
(x_8^{(r+1)}, x_9^{(r+1)}, x_{10}^{(r+1)}, x_{11}^{(r+1)}) &\leftarrow T_0[x_8^{(r)}] \oplus T_1[x_{13}^{(r)}] \oplus T_2[x_2^{(r)}] \oplus T_3[x_7^{(r)}] \oplus K_2^{(r+1)} \\
(x_{12}^{(r+1)}, x_{13}^{(r+1)}, x_{14}^{(r+1)}, x_{15}^{(r+1)}) &\leftarrow T_0[x_{12}^{(r)}] \oplus T_1[x_1^{(r)}] \oplus T_2[x_6^{(r)}] \oplus T_3[x_{11}^{(r)}] \oplus K_3^{(r+1)}
\end{aligned} \tag{1}$$

Finally, to compute the last round (1) is repeated with $r = 9$, except that $T_0, \ldots, T_3$ is replaced by $T_0^{(10)}, \ldots, T_3^{(10)}$. The resulting $\boldsymbol{x}^{(10)}$ is the ciphertext. Compared to the algebraic formulation of AES, here the lookup tables account for the combination of ShiftRows, MixColumns and SubBytes operations; the change of lookup tables for the last is due to the absence of MixColumns.

[3] Some implementations use variant with a different table layouts; see Section 5.2.

[4] We consider AES with 128-bit keys. The attacks can be adapted to longer keys.

2.3 Notation

We treat bytes interchangeably as integers in $\{0, \ldots, 255\}$ and as elements of $\{0,1\}^8$ that can be XORed. Let δ denote the cache line size B divided by the size of each table entry (usually 4 bytes); on most platforms of interest we have $\delta = 16$. For a byte y and table T_ℓ, we will denote $\langle y \rangle = \lfloor y/\delta \rfloor$ and call this *the memory block of* y in T_ℓ. The significance of this notation is as follows: two bytes y, z fulfill $\langle y \rangle = \langle z \rangle$ iff, when used as lookup indices into the same table T_ℓ, they would cause access to the same memory block[5]; they would therefore be impossible to distinguish based only on a single memory access. For a byte y and table T_ℓ, we say that an AES encryption with given inputs *accesses the memory block of* y *in* T_ℓ if, according to the above description of AES, at some point in the encryption there will be some table lookup to $T_\ell[z]$ where $\langle z \rangle = \langle y \rangle$.

In Section 3 we will show methods for discovering (and taking advantage of the discovery) whether the encryption code, invoked as a black box, accesses a given memory block. To this end we define the following predicate: $Q_{\boldsymbol{k}}(\boldsymbol{p}, \ell, y) = 1$ iff the AES encryption of the plaintext $\boldsymbol{p}$ under the encryption key $\boldsymbol{k}$ accesses the memory block of index y in T_ℓ at least once throughout the 10 rounds.

Also in Section 3, our measurement procedures will sample *measurement score* from a distribution $M_{\boldsymbol{k}}(\boldsymbol{p}, \ell, y)$ over $\mathbb{R}$. The exact definition of $M_{\boldsymbol{k}}(\boldsymbol{p}, \ell, y)$ will vary, but it will approximate $Q_{\boldsymbol{k}}(\boldsymbol{p}, \ell, y)$ in the following rough sense: for a large fraction of the keys $\boldsymbol{k}$, all tables ℓ and a large fraction of the indices $\boldsymbol{x}$, for random plaintexts and measurement noise, the expectation of $M_{\boldsymbol{k}}(\boldsymbol{p}, \ell, y)$ is larger when $Q_{\boldsymbol{k}}(\boldsymbol{p}, \ell, y) = 1$ than when $Q_{\boldsymbol{k}}(\boldsymbol{p}, \ell, y) = 0$.

3 Synchronous Known-Data Attacks

3.1 Overview

The first family of attacks, termed *synchronous attacks*, is applicable in scenarios where the plaintext or ciphertext is known and the attacker can operate synchronously with the encryption on the same processor, by using (or eavesdropping upon) some interface that triggers encryption under an unknown key. For example, a Virtual Private Network may allow an unprivileged user to send data packets through a secure channel. This lets the user trigger encryption of plaintexts that are mostly known (up to some uncertainties in the packet headers), and our attack would thus, under some circumstances, enable any such user to discover the key used by the VPN to protect all users' packets. As another example, consider the Linux `dm-crypt` and `cryptoloop` services. These allow the administrator to create a virtual device which provides encrypted storage into an

[5] We assume that the tables are aligned on memory block boundaries, which is usually the case. Non-aligned tables would *benefit* our attacks by leaking an extra bit per key byte in the first round. We also assume for simplicity that all tables are mapped into distinct cache sets; this holds with high probability on many systems (and our practical attacks also handle some exceptions).