Alex Biryukov (Ed.)

LNCS 4593

Fast Software Encryption

14th International Workshop, FSE 2007
Luxembourg, Luxembourg, March 2007
Revised Selected Papers

Springer

Alex Biryukov (Ed.)

Fast Software Encryption

14th International Workshop, FSE 2007
Luxembourg, Luxembourg, March 26-28, 2007
Revised Selected Papers

Springer

Volume Editor

Alex Biryukov
FSTC, University of Luxembourg
6, rue Richard Coudenhove-Kalergi, 1359 Luxembourg-Kirchberg, Luxembourg
E-mail: alex.biryukov@uni.lu

Library of Congress Control Number: 2007933305

CR Subject Classification (1998): E.3, F.2.1, E.4, G.2, G.4

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-540-74617-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-74617-1 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12115600 06/3180 5 4 3 2 1 0

Lecture Notes in Computer Science 4593

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Preface

Fast Software Encryption 2007 was the 14th annual workshop in the series, which was sponsored by the International Association for Cryptologic Research (IACR) for the sixth time. FSE has become a brand which attracts top research papers on symmetric cryptology. This includes papers on fast and secure primitives for symmetric cryptography, such as the design and analysis of block ciphers, stream ciphers, encryption schemes, hash functions, and message authentication codes (MACs), and on tools for analysis and evaluation. Previous editions of FSE took place in Cambridge, Leuven, Haifa, Rome, New York, Yokohama, Lund, Delhi, Paris, and Graz.

The Fast Software Encryption 2007 workshop was held March 26-28, 2007 in Luxembourg. It was organized by the General Chair Jean-Claude Asselborn (University of Luxembourg) in cooperation with the research lab LACS (Laboratory of Algorithms, Cryptography and Security) of the Computer Science and Communications research unit of the University of Luxembourg. The conference was attended by 160 registered participants from 36 different countries.

There were 104 papers submitted to FSE 2007, from which 28 were selected for presentation. The selection of papers was a challenging task, each submission had at least four reviewers, papers from Program Committee members having at least five. About 450 reviews were written by the committee and the external reviewers. The discussion phase was very fruitful, leading to more than 400 discussion comments in total, with several discussions going beyond 20 comments. I would like to thank the Program Committee and the external reviewers, who did an excellent job. It was a real pleasure to work with this team.

The conference program also featured an invited talk by Jean-Charles Faugére on the topic "Groebner Bases. Applications in Cryptology." The traditional rump session with short informal presentations of recent results was chaired by Joan Daemen.

We would also like to thank the following people: Thomas Baignères and Matthieu Finiasz as the authors of the iChair review software; Dmitry Khovratovich for his help with the conference Web site and compilation of the proceedings; Volker Müller, Michel Carpentier, Christian Hutter, and SIU for videotaping the talks and providing a wireless LAN for the participants. We would like to thank the students of the Lycée Technique "Ecole de Commerce et de Gestion" and our secretaries Elisa Ferreira, Ragga Eyjolfsdottir, and Mireille Kies for their help in the organization of the workshop. We would also like to thank IACR and in particular Helena Handschuh, Shai Halevi, and Bart Preneel for constant support. Thanks to Britta Schlüter for the public relations work. Finally we are grateful to our sponsors FNR — Luxembourg National Research Fund — and the University of Luxembourg as well as the Centre de Culture et de Rencontre Neumünster, Ministry of Culture, Research and Universities.

March 2007 Alex Biryukov

FSE 2007

March 26–28, 2007, Luxembourg City, Luxembourg

Sponsored by
the International Association for Cryptologic Research (IACR)

General Chair

Jean-Claude Asselborn, University of Luxembourg, Luxembourg

Program Chair

Alex Biryukov, University of Luxembourg, Luxembourg

Program Committee

Frederik Armknecht	NEC, Germany
Steve Babbage	Vodafone, UK
Alex Biryukov (chair)	University of Luxembourg, Luxembourg
Claude Carlet	University of Paris 8 and INRIA, France
Nicolas Courtois	University College London, UK
Joan Daemen	STMicroelectronics, Belgium
Orr Dunkelman	K.U.Leuven, Belgium
Henri Gilbert	France Telecom, France
Louis Granboulan	EADS, France
Helena Handschuh	Spansion, France
Jin Hong	Seoul National University, Korea
Seokhie Hong	CIST, Korea
Tetsu Iwata	Nagoya University, Japan
Thomas Johansson	Lund University, Sweden
Antoine Joux	DGA and University of Versailles, France
Pascal Junod	Nagravision, Switzerland
Charanjit Jutla	IBM T.J. Watson Research Center, USA
John Kelsey	NIST, USA
Lars R. Knudsen	Technical University of Denmark, Denmark
Stefan Lucks	University of Mannheim, Germany
Mitsuru Matsui	Mitsubishi Electric, Japan
Willi Meier	FHNW, Switzerland

Kaisa Nyberg	Nokia and Helsinki University of Technology, Finland
Elisabeth Oswald	University of Bristol, UK
Josef Pieprzyk	Macquarie University, Australia
Bart Preneel	K.U.Leuven, Belgium
Greg Rose	Qualcomm, USA
Palash Sarkar	Indian Statistical Institute, India
Serge Vaudenay	EPFL, Switzerland

Subreviewers

Elena Andreeva
Thomas Baignères
Gregory V. Bard
Côme Berbain
Guido Bertoni
Olivier Billet
Nick Bone
Christophe De Cannière
Chris Charnes
Lily Chen
Scott Contini
Morris Dworkin
Martin Feldhofer
Matthieu Finiasz
Benedikt Gierlichs
Sylvain Guilley
Philip Hawkes
Christoph Herbst
Katrin Hoeper
Deukjo Hong
Alexandre Karlov
Nathan Keller
Alexander Kholosha
Dmitry Khovratovich
Jongsung Kim
Andrew Klapper
Özgül Küçük
Ulrich Kühn
Changhoon Lee
Svetla Nikova
Stefan Mangard
Stéphane Manuel
Krystian Matusiewicz
Alexander Maximov
Cameron McDonald
Florian Mendel
Marine Minier
Joydip Mitra
Jean Monnerat
Alp Öztarhan
Sylvain Pasini
Ludovic Perret
Thomas Peyrin
Gilles Piret
Thomas Popp
Norbert Pramstaller
Emmanuel Prouff
Christian Rechberger
Matt Robshaw
Allen Roginsky
Martin Schläffer
Yannick Seurin
Nicolas Sendrier
Igor Shparlinski
Soren Steffen Thomsen
Dirk Stegemann
Ron Steinfeld
Jaechul Sung
Daisuke Suzuki
Emin Tatli
Charlotte Vikkelsoe
Martin Vuagnoux
Ralf-Philipp Weinmann
Christopher Wolf
Hongjun Wu
Jin Yuan
Erik Zenner

Lecture Notes in Computer Science

For information about Vols. 1–4570

please contact your bookseller or Springer

Vol. 4708: L. Kučera, A. Kučera (Eds.), Mathematical Foundations of Computer Science 2007. XVIII, 764 pages. 2007.

Vol. 4707: O. Gervasi, M.L. Gavrilova (Eds.), Computational Science and Its Applications – ICCSA 2007, Part III. XXIV, 1205 pages. 2007.

Vol. 4706: O. Gervasi, M.L. Gavrilova (Eds.), Computational Science and Its Applications – ICCSA 2007, Part II. XXIII, 1129 pages. 2007.

Vol. 4705: O. Gervasi, M.L. Gavrilova (Eds.), Computational Science and Its Applications – ICCSA 2007, Part I. XLIV, 1169 pages. 2007.

Vol. 4703: L. Caires, V.T. Vasconcelos (Eds.), CONCUR 2007 – Concurrency Theory. XIII, 507 pages. 2007.

Vol. 4697: L. Choi, Y. Paek, S. Cho (Eds.), Advances in Computer Systems Architecture. XIII, 400 pages. 2007.

Vol. 4685: D.J. Veit, J. Altmann (Eds.), Grid Economics and Business Models. XII, 201 pages. 2007.

Vol. 4683: L. Kang, Y. Liu, S. Zeng (Eds.), Intelligence Computation and Applications. XVII, 663 pages. 2007.

Vol. 4682: D.-S. Huang, L. Heutte, M. Loog (Eds.), Advanced Intelligent Computing Theories and Applications. XXVII, 1373 pages. 2007. (Sublibrary LNAI).

Vol. 4681: D.-S. Huang, L. Heutte, M. Loog (Eds.), Advanced Intelligent Computing Theories and Applications. XXVI, 1379 pages. 2007.

Vol. 4679: A.L. Yuille, S.-C. Zhu, D. Cremers, Y. Wang (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition. XII, 494 pages. 2007.

Vol. 4673: W.G. Kropatsch, M. Kampel, A. Hanbury (Eds.), Computer Analysis of Images and Patterns. XX, 1006 pages. 2007.

Vol. 4671: V. Malyshkin (Ed.), Parallel Computing Technologies. XIV, 635 pages. 2007.

Vol. 4660: S. Džeroski, J. Todorovski (Eds.), Computational Discovery of Scientific Knowledge. X, 327 pages. 2007. (Sublibrary LNAI).

Vol. 4657: C. Lambrinoudakis, G. Pernul, A M. Tjoa (Eds.), Trust and Privacy in Digital Business. XIII, 291 pages. 2007.

Vol. 4651: F. Azevedo, P. Barahona, F. Fages, F. Rossi (Eds.), Recent Advances in Constraints. VIII, 185 pages. 2007. (Sublibrary LNAI).

Vol. 4649: V. Diekert, M.V. Volkov, A. Voronkov (Eds.), Computer Science – Theory and Applications. XIII, 420 pages. 2007.

Vol. 4647: R. Martin, M. Sabin, J. Winkler (Eds.), Mathematics of Surfaces XII. IX, 509 pages. 2007.

Vol. 4645: R. Giancarlo, S. Hannenhalli (Eds.), Algorithms in Bioinformatics. XIII, 432 pages. 2007. (Sublibrary LNBI).

Vol. 4644: N. Azemard, L. Svensson (Eds.), Integrated Circuit and System Design. XIV, 583 pages. 2007.

Vol. 4643: M.-F. Sagot, M.E.M.T. Walter (Eds.), Advances in Bioinformatics and Computational Biology. XII, 177 pages. 2007. (Sublibrary LNBI).

Vol. 4642: S.-W. Lee, S.Z. Li (Eds.), Advances in Biometrics. XX, 1216 pages. 2007.

Vol. 4639: E. Csuhaj-Varj'u, Z. Ésik (Eds.), Fundamentals of Computation Theory. XIV, 508 pages. 2007.

Vol. 4638: T. Stützle, M. Birattari, H.H. Hoos (Eds.), Engineering Stochastic Local Search Algorithms. X, 223 pages. 2007.

Vol. 4637: C. Kruegel, R. Lippmann, A. Clark (Eds.), Recent Advances in Intrusion Detection. XII, 337 pages. 2007.

Vol. 4635: B. Kokinov, D.C. Richardson, T.R. Roth-Berghofer, L. Vieu (Eds.), Modeling and Using Context. XIV, 574 pages. 2007. (Sublibrary LNAI).

Vol. 4634: H.R. Nielson, G. Filé (Eds.), Static Analysis. XI, 469 pages. 2007.

Vol. 4633: M. Kamel, A. Campilho (Eds.), Image Analysis and Recognition. XII, 1312 pages. 2007.

Vol. 4632: R. Alhajj, H. Gao, X. Li, J. Li, O.R. Zaïane (Eds.), Advanced Data Mining and Applications. XV, 634 pages. 2007. (Sublibrary LNAI).

Vol. 4628: L.N. de Castro, F.J. Von Zuben, H. Knidel (Eds.), Artificial Immune Systems. XII, 438 pages. 2007.

Vol. 4627: M. Charikar, K. Jansen, O. Reingold, J.D.P. Rolim (Eds.), Approximation, Randomization, and Combinatorial Optimization. XII, 626 pages. 2007.

Vol. 4626: R.O. Weber, M.M. Richter (Eds.), Case-Based Reasoning Research and Development. XIII, 534 pages. 2007. (Sublibrary LNAI).

Vol. 4624: T. Mossakowski, U. Montanari, M. Haveraaen (Eds.), Algebra and Coalgebra in Computer Science. XI, 463 pages. 2007.

Vol. 4622: A. Menezes (Ed.), Advances in Cryptology - CRYPTO 2007. XIV, 631 pages. 2007.

Vol. 4619: F. Dehne, J.-R. Sack, N. Zeh (Eds.), Algorithms and Data Structures. XVI, 662 pages. 2007.

Vol. 4618: S.G. Akl, C.S. Calude, M.J. Dinneen, G. Rozenberg, H.T. Wareham (Eds.), Unconventional Computation. X, 243 pages. 2007.

Vol. 4617: V. Torra, Y. Narukawa, Y. Yoshida (Eds.), Modeling Decisions for Artificial Intelligence. XII, 502 pages. 2007. (Sublibrary LNAI).

Vol. 4616: A. Dress, Y. Xu, B. Zhu (Eds.), Combinatorial Optimization and Applications. XI, 390 pages. 2007.

Vol. 4615: R. de Lemos, C. Gacek, A. Romanovsky (Eds.), Architecting Dependable Systems IV. XIV, 435 pages. 2007.

Vol. 4613: F.P. Preparata, Q. Fang (Eds.), Frontiers in Algorithmics. XI, 348 pages. 2007.

Vol. 4612: I. Miguel, W. Ruml (Eds.), Abstraction, Reformulation, and Approximation. XI, 418 pages. 2007. (Sublibrary LNAI).

Vol. 4611: J. Indulska, J. Ma, L.T. Yang, T. Ungerer, J. Cao (Eds.), Ubiquitous Intelligence and Computing. XXIII, 1257 pages. 2007.

Vol. 4610: B. Xiao, L.T. Yang, J. Ma, C. Muller-Schloer, Y. Hua (Eds.), Autonomic and Trusted Computing. XVIII, 571 pages. 2007.

Vol. 4609: E. Ernst (Ed.), ECOOP 2007 – Object-Oriented Programming. XIII, 625 pages. 2007.

Vol. 4608: H.W. Schmidt, I. Crnkovic, G.T. Heineman, J.A. Stafford (Eds.), Component-Based Software Engineering. XII, 283 pages. 2007.

Vol. 4607: L. Baresi, P. Fraternali, G.-J. Houben (Eds.), Web Engineering. XVI, 576 pages. 2007.

Vol. 4606: A. Pras, M. van Sinderen (Eds.), Dependable and Adaptable Networks and Services. XIV, 149 pages. 2007.

Vol. 4605: D. Papadias, D. Zhang, G. Kollios (Eds.), Advances in Spatial and Temporal Databases. X, 479 pages. 2007.

Vol. 4604: U. Priss, S. Polovina, R. Hill (Eds.), Conceptual Structures: Knowledge Architectures for Smart Applications. XII, 514 pages. 2007. (Sublibrary LNAI).

Vol. 4603: F. Pfenning (Ed.), Automated Deduction – CADE-21. XII, 522 pages. 2007. (Sublibrary LNAI).

Vol. 4602: S. Barker, G.-J. Ahn (Eds.), Data and Applications Security XXI. X, 291 pages. 2007.

Vol. 4600: H. Comon-Lundh, C. Kirchner, H. Kirchner (Eds.), Rewriting, Computation and Proof. XVI, 273 pages. 2007.

Vol. 4599: S. Vassiliadis, M. Berekovic, T.D. Hämäläinen (Eds.), Embedded Computer Systems: Architectures, Modeling, and Simulation. XVIII, 466 pages. 2007.

Vol. 4598: G. Lin (Ed.), Computing and Combinatorics. XII, 570 pages. 2007.

Vol. 4597: P. Perner (Ed.), Advances in Data Mining. XI, 353 pages. 2007. (Sublibrary LNAI).

Vol. 4596: L. Arge, C. Cachin, T. Jurdziński, A. Tarlecki (Eds.), Automata, Languages and Programming. XVII, 953 pages. 2007.

Vol. 4595: D. Bošnački, S. Edelkamp (Eds.), Model Checking Software. X, 285 pages. 2007.

Vol. 4594: R. Bellazzi, A. Abu-Hanna, J. Hunter (Eds.), Artificial Intelligence in Medicine. XVI, 509 pages. 2007. (Sublibrary LNAI).

Vol. 4593: A. Biryukov (Ed.), Fast Software Encryption. XI, 467 pages. 2007.

Vol. 4592: Z. Kedad, N. Lammari, E. Métais, F. Meziane, Y. Rezgui (Eds.), Natural Language Processing and Information Systems. XIV, 442 pages. 2007.

Vol. 4591: J. Davies, J. Gibbons (Eds.), Integrated Formal Methods. IX, 660 pages. 2007.

Vol. 4590: W. Damm, H. Hermanns (Eds.), Computer Aided Verification. XV, 562 pages. 2007.

Vol. 4589: J. Münch, P. Abrahamsson (Eds.), Product-Focused Software Process Improvement. XII, 414 pages. 2007.

Vol. 4588: T. Harju, J. Karhumäki, A. Lepistö (Eds.), Developments in Language Theory. XI, 423 pages. 2007.

Vol. 4587: R. Cooper, J. Kennedy (Eds.), Data Management. XIII, 259 pages. 2007.

Vol. 4586: J. Pieprzyk, H. Ghodosi, E. Dawson (Eds.), Information Security and Privacy. XIV, 476 pages. 2007.

Vol. 4585: M. Kryszkiewicz, J.F. Peters, H. Rybinski, A. Skowron (Eds.), Rough Sets and Intelligent Systems Paradigms. XIX, 836 pages. 2007. (Sublibrary LNAI).

Vol. 4584: N. Karssemeijer, B. Lelieveldt (Eds.), Information Processing in Medical Imaging. XX, 777 pages. 2007.

Vol. 4583: S.R. Della Rocca (Ed.), Typed Lambda Calculi and Applications. X, 397 pages. 2007.

Vol. 4582: J. Lopez, P. Samarati, J.L. Ferrer (Eds.), Public Key Infrastructure. XI, 375 pages. 2007.

Vol. 4581: A. Petrenko, M. Veanes, J. Tretmans, W. Grieskamp (Eds.), Testing of Software and Communicating Systems. XII, 379 pages. 2007.

Vol. 4580: B. Ma, K. Zhang (Eds.), Combinatorial Pattern Matching. XII, 366 pages. 2007.

Vol. 4579: B. M. Hämmerli, R. Sommer (Eds.), Detection of Intrusions and Malware, and Vulnerability Assessment. X, 251 pages. 2007.

Vol. 4578: F. Masulli, S. Mitra, G. Pasi (Eds.), Applications of Fuzzy Sets Theory. XVIII, 693 pages. 2007. (Sublibrary LNAI).

Vol. 4577: N. Sebe, Y. Liu, Y.-t. Zhuang, T.S. Huang (Eds.), Multimedia Content Analysis and Mining. XIII, 513 pages. 2007.

Vol. 4576: D. Leivant, R. de Queiroz (Eds.), Logic, Language, Information and Computation. X, 363 pages. 2007.

Vol. 4575: T. Takagi, T. Okamoto, E. Okamoto, T. Okamoto (Eds.), Pairing-Based Cryptography – Pairing 2007. XI, 408 pages. 2007.

Vol. 4574: J. Derrick, J. Vain (Eds.), Formal Techniques for Networked and Distributed Systems – FORTE 2007. XI, 375 pages. 2007.

Vol. 4573: M. Kauers, M. Kerber, R. Miner, W. Windsteiger (Eds.), Towards Mechanized Mathematical Assistants. XIII, 407 pages. 2007. (Sublibrary LNAI).

Vol. 4572: F. Stajano, C. Meadows, S. Capkun, T. Moore (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. X, 247 pages. 2007.

Vol. 4571: P. Perner (Ed.), Machine Learning and Data Mining in Pattern Recognition. XIV, 913 pages. 2007. (Sublibrary LNAI).

Table of Contents

Hash Function Cryptanalysis and Design (I)

Stream Ciphers Cryptanalysis (I)

Theory

Fast Talks: Block Cipher Cryptanalysis

Fast Talks: Block Cipher Design

Block Cipher Cryptanalysis

Stream Cipher Cryptanalysis (II)

Cryptanalysis of Hash Functions (II)

Theory of Stream Ciphers

Side Channel Attacks

MACs and Small Block Ciphers

Producing Collisions for PANAMA, Instantaneously

Joan Daemen and Gilles Van Assche

STMicroelectronics, Zaventem, Belgium
gro.noekeon@noekeon.org

Abstract. We present a practical attack on the PANAMA hash function that generates a collision in 2^6 evaluations of the state updating function. Our attack improves that of Rijmen and coworkers that had a complexity 2^{82}, too high to produce a collision in practice. This improvement comes mainly from the use of techniques to transfer conditions on the state to message words instead of trying many message pairs and using the ones for which the conditions are satisfied. Our attack works for any arbitrary prefix message, followed by a pair of suffix messages with a given difference. We give an example of a collision and make the collision-generating program available. Our attack does not affect the PANAMA stream cipher, that is still unbroken to the best of our knowledge.

Keywords: symmetric cryptography, hash function, collision.

1 Introduction

A cryptographic hash function maps a message of arbitrary length to a fixed-size output called a digest. One of the requirements for a cryptographic hash function is collision-resistance: it should be infeasible to to find two different messages that give the same digest.

PANAMA can be used both as a hash function and as a stream cipher. In the scope of this paper, we will consider only on the hash part. Our attack does not have impact on the security of the PANAMA stream cipher. Internally, PANAMA has a state and a buffer, which evolve using a state updating function. For every block of message, the state updating function transforms the state and the buffer. We describe PANAMA in Sec. 2.

In this article, we describe a method to produce collisions for the PANAMA hash function that refines the method of Rijmen and coworkers [2] and reduces the workload from 2^{82} to 2^6 applications of the state updating function. We can therefore generate collisions quasi instantaneously. Furthermore, there are many degrees of freedom in the produced messages. The attack works for any initial value of the state. This means that one can find a collision with a pair of messages $(M_1|M, M_1|M^*)$ with an arbitrary prefix M_1. Here, the message parts M and M^* have a fixed difference $M' = M + M^*$. Furthermore, the attacker

A. Biryukov (Ed.): FSE 2007, LNCS 4593, pp. 1–18, 2007.

can append an arbitrary suffix M_2 to both collision messages, independently of M_1, M and M^*. We discuss the structure of the attack in Sec. 3.

Like in [2], we use a differential trail (called differential path in [2]) that leads to a zero difference in state and buffer. A differential trail specifies both the message differences and the differences in the state and in the buffer. For a pair of messages to follow the right differences in the state, a subset of the state bits must satisfy specific conditions. In the attack in [2], part of these conditions were transferred to equations on message words while the remaining ones were satisfied by trying many different message pairs and picking out those for which these conditions happened to be satisfied. In our attack, we transfer *all* conditions to equations on message bits using some simple new techniques explained in Sec. 4. The transfer of equations has negligible workload.

Although very similar, our trail is different from that of [2]. We chose a trail such that the conditions on the state are more easily transferable to equations on the message bits. We describe it in Sec. 5 and all its conditions and their transfer in Sec. 6.

2 Description of PANAMA

The internal memory of PANAMA is composed of 273 32-bit words (hereby denoted words) and is organized in two parts: [1]

- the *state*, with 17 words denoted a_0 through a_{16}, and
- the *buffer*, which is an array of 32×8 words, denoted $b_{i,j}$ with $0 \leq i \leq 31$ and $0 \leq j \leq 7$. (Note that b_i indicates a block of 8 words $b_{i,0} \ldots b_{i,7}$.)

The + sign applied on bits denotes the exclusive *or* (xor) operation and on words the bitwise xor. In subscripts of the state a, it denotes modulo-17 addition.

The message to hash is padded and divided into blocks of 8 words (i.e., 256 bits) each. It is processed as follows. First, both the state and the buffer are initialized to 0. Then, for each message block $p = (p_0, p_1, \ldots, p_7)$ (i.e., for each round), the following operations are applied:

- the state undergoes a non-linear transformation $\theta \circ \pi \circ \gamma$, with

$$\begin{aligned} \gamma &: a_i \leftarrow a_i + (a_{i+1} + \overline{0})a_{i+2} + \overline{0}, \\ \pi &: a_i \leftarrow a_{7i \bmod 17} \ggg i(i+1)/2, \\ \theta &: a_i \leftarrow a_i + a_{i+1} + a_{i+4}, \end{aligned}$$

  where the invisible multiplication indicates the bitwise *and*, $\overline{0}$ denotes the word with 32 bits 1, and $\ggg$ cyclic right shift of the bits within a word;
- the least significant bit of a_0 is flipped: $a_0 \leftarrow a_0 + 1$;
- the message block is xored into the state:

$$a \leftarrow a + f_{\text{i}\rightarrow\text{s}}(p) \;\Leftrightarrow\; a_{i+1} \leftarrow a_{i+1} + p_i,\ 0 \leq i \leq 7;$$

– eight words of the buffer are xored into the state:

$$a \leftarrow a + f_{\mathrm{b}\to\mathrm{s}}(b_{16}) \;\Leftrightarrow\; a_{i+9} \leftarrow a_{i+9} + b_{16,i},\ 0 \le i \le 7;$$

– the buffer undergoes a linear feedback shift register (LFSR) step:

$$\begin{aligned} b_i &\leftarrow b_{i-1 \bmod 32} \quad (i \neq 25), \\ b_{25} &\leftarrow b_{24} + r(b_{31}), \end{aligned}$$

where the function r is defined as $Y = r(X) \Leftrightarrow Y_j = X_{j+2 \bmod 8}$;

– the message block is xored into the buffer: $b_{0,i} \leftarrow b_{0,i} + p_i$, $0 \le i \le 7$.

After all the message blocks are processed, 33 extra rounds are performed, called blank rounds. These rounds use the state updating function, with the difference that a part of the state (instead of a message block) is input into the buffer: $b_{0,i} \leftarrow b_{0,i} + a_{i+1}$, $0 \le i \le 7$.

Finally, the digest is extracted from the state after the blank rounds.

3 Structure of the Attack

The first thing to note is that the presence of the blank rounds makes it hard to produce a collision in the digest if there is a difference in either the state or the buffer after all the message blocks are input. Due to the invertibility of the state updating function such a difference will not cancel out. Moreover, the lack of external input and the propagation properties of the state updating function give the attacker almost no control over the final difference. Therefore, our goal is to produce a collision in both the state and the buffer before the blank rounds.

We produce a collision by following a trail. Two instances of PANAMA process two different messages (p and $p + dp$), which have a given difference (dp). The trail also specifies the differences in the state (da) and in the buffer (db) between the two instances of PANAMA, at each round. So, not only the two messages must have the given difference, they must also produce the right difference in the state and in the buffer.

We shall now describe the general structure of the trail used in the scope of this article. We will first talk about the sequence of message differences, then about the differences in the state.

In the sequel, the round numbers are specified between brackets in superscript: $.^{(i)}$. The convention is that $p^{(i)}$ is the message block processed during round i, and $a^{(i)}$ is the value of the state after round i.

3.1 Collision in the Buffer

The buffer evolves independently from the state and is linear. As noticed in [2], the following message difference sequence gives a collision in the buffer for any x:

$$dp^{(1)} = x,\ dp^{(8)} = r(x),\ dp^{(33)} = x, \text{ all other differences } 0. \tag{1}$$

After 32 rounds, we have $db_{24} = r(x)$ and $db_{31} = x$. After the 33rd round, we get:

$$db_{25} \leftarrow db_{24} + r(db_{31}) = r(x) + r(x) = 0,$$
$$db_0 \leftarrow db_{31} + dp^{(33)} = x + x = 0.$$

Thanks to the linearity of the buffer, any combination of shifted instances of the sequence (1) results in a collision in the buffer. In [2], two such sequences are used, one distant of two rounds from the other. In this paper, we instead use three such sequences at three consecutive rounds. More precisely, the message sequence is as follows (only non-zero differences are indicated):

$$\begin{aligned}(dp^{(1)}, dp^{(2)}, dp^{(3)}) &= (d^{(1)}, d^{(2)}, d^{(3)}),\\ (dp^{(8)}, dp^{(9)}, dp^{(10)}) &= (r(d^{(1)}), r(d^{(2)}), r(d^{(3)}),\\ (dp^{(33)}, dp^{(34)}, dp^{(35)}) &= (d^{(1)}, d^{(2)}, d^{(3)}).\end{aligned}$$

3.2 Collision in the State

The state is influenced both by the message blocks and by the buffer words in b_{16}. Let us summarize the sequence of differences that are xored into the state, both from the message block and from b_{16}:

I Rounds $r = i + 0$: State gets difference $dp^{(r)} = d^{(i)}$
II Rounds $r = i + 7$: State gets difference $dp^{(r)} = r(d^{(i)})$
III Rounds $r = i + 17$: State gets difference $db_{16}^{(r)} = dp^{(r-17)} = d^{(i)}$
IV Rounds $r = i + 24$: State gets difference $db_{16}^{(r)} = dp^{(r-17)} = r(d^{(i)})$
V Rounds $r = i + 32$: State gets difference $dp^{(r)} = d^{(i)}$

with $1 \leq i \leq 3$.

After the three rounds in each of the five sequences described above, we will make sure that we have a collision in the state. These are called *subcollisions*. After the last subcollision, we have both a collision in the state and in the buffer, and we are thus guaranteed to obtain the same digest after the blank rounds.

Before we explain how to obtain a subcollision, we need to detail the properties of the difference propagation in γ, the only non-linear operation of the state updating function.

3.3 Difference Propagation Through γ

Since γ is composed only of bitwise operations, we will only talk about γ as if it operates on 17 bits in this current subsection. The actual γ on words can be seen as 32 such operations in parallel.

Assume that the input of one instance of γ is a, while the input of the other instance is $a + da$. For a given input difference da, not all output differences are possible. The output difference $dc = (dc_0, \ldots, dc_{16})$ is determined by the following equation:

$$dc_i = \gamma_i(da) + da_{i+1}a_{i+2} + da_{i+2}a_{i+1} + 1,$$

where $\gamma_i(a) = a_i + (a_{i+1} + 1)a_{i+2} + 1$ denotes a particular output bit of γ.

Hence, we can obtain an output difference dc from a given input difference da only if a satisfies some conditions. These are as follows:

$$\text{If } da_{i+1} = 1 \text{ and } da_{i+2} = 0, \text{ then } a_{i+2} = dc_i + \gamma_i(da) + 1; \quad (2)$$

$$\text{If } da_{i+1} = 0 \text{ and } da_{i+2} = 1, \text{ then } a_{i+1} = dc_i + \gamma_i(da) + 1; \quad (3)$$

$$\text{If } da_{i+1} = 1 \text{ and } da_{i+2} = 1, \text{ then } a_{i+1} + a_{i+2} = dc_i + \gamma_i(da) + 1. \quad (4)$$

We call conditions of type (2) and (3) *simple* conditions and conditions of type (4) *two-bit parity* conditions. We call a differential (da, dc) for which the set of conditions has a solution a *possible differential*.

Note that the input difference da fully determines the positions of the state bits a_i that are subject to conditions. Assume that we have n consecutive 1s in the pattern da, i.e., we have $da_i = da_{i+n+1} = 0$ and in between $da_{i+l} = 1$ $(1 \leq l \leq n)$. Then there are simple conditions on a_i and on a_{i+n+1}, and $n-1$ two-bit parity conditions on $a_{i+l} + a_{i+l+1}$ $(1 \leq l < n)$. This can be applied to all such patterns in da.

From this follows that the number of conditions is equal to the Hamming weight of da plus the number of 001 patterns in da. (For the particular case of da = 1111111111111111, there are 16 independent two-bit parity conditions.) We denote by $w(da)$ the number of conditions due to da.

3.4 Specifying the Trail

For our attack to work, we wish to determine equations on the message bits that imply the five subcollisions. In the previous subsection we have shown that given a possible differential (da, dc) over γ, we obtain conditions on input bits of γ.

Consider now subcollision I. Before the first round, there is no difference in the state, hence $da^{(0)} = 0$. At the input of the second round, the message difference appears in the state: $da^{(1)} = f_{\text{i}\to\text{s}}(d^{(1)})$. This determines the input difference of γ in round 2. We now need to specify the output of γ in the second round, but we can equivalently specify $da^{(2)}$, as the other operations are linear. After the third round, the fact that we have a collision in the state imposes that $da^{(3)} = 0$, yielding at the output of the third round a difference equal to $f_{\text{i}\to\text{s}}(d^{(3)})$. Hence a value for $da^{(2)}$ must be chosen such that differentials $(f_{\text{i}\to\text{s}}(d^{(1)}), \pi^{-1} \circ \theta^{-1}(da^{(2)} + f_{\text{i}\to\text{s}}(d^{(2)})))$ and $(da^{(2)}, \pi^{-1} \circ \theta^{-1}(f_{\text{i}\to\text{s}}(d^{(3)})))$ over γ are possible. For a given message difference sequence $d^{(1)}, d^{(2)}, d^{(3)}$ there may be several, one or none such values of $da^{(2)}$. Note that the first differential imposes conditions on $a^{(1)}$ and the second one on $a^{(2)}$.

As θ and π are linear, it follows that a possible differential over the state-updating function imposes conditions on bits of the state $a^{(i)}$. Doing this for differentials over more rounds is more difficult and we avoid it in out attack. Therefore, for each round in which there is non-zero input difference in the state, we need to know the output difference.

For subcollisions II to V, applying the same reasoning leads to following round differentials, which we write as differentials over $\theta \circ \pi \circ \gamma$ for compactness: