Serge Vaudenay
Amr M. Youssef (Eds.)

LNCS 2259

Selected Areas in Cryptography

8th Annual International Workshop, SAC 2001
Toronto, Ontario, Canada, August 2001
Revised Papers

Springer

Serge Vaudenay Amr M. Youssef (Eds.)

Selected Areas in Cryptography

8th Annual International Workshop, SAC 2001
Toronto, Ontario, Canada, August 16-17, 2001
Revised Papers

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Serge Vaudenay
EPFL, LASEC
1015 Lausanne, Switzerland
E-mail: serge.vaudenay@epfl.ch

Amr M. Youssef
University of Waterloo, CACR
Waterloo, Ontario N2L 3G1, Canada
E-mail: a2youssef@cacr.math.uwaterloo.ca

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Selected areas in cryptography : 8th annual international workshop ; revised papers / SAC 2001, Toronto, Ontario, Canada, August 16 - 17, 2001. Serge Vaudenay ; Amr M. Youssef (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2259)
ISBN 3-540-43066-0

CR Subject Classification (1998): E.3, D.4.6, K.6.5, F.2.1-2, C.2, H.4.3

ISSN 0302-9743
ISBN 3-540-43066-0 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Steingräber Satztechnik GmbH, Heidelberg
Printed on acid-free paper SPIN: 10846068 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science

For information about Vols. 1–2175
please contact your bookseller or Springer-Verlag

Vol. 2128: H. Ehrig, G. Juhás, J. Padberg, G. Rozenberg (Eds.), Unifying Petri Nets. VIII, 485 pages. 2001.

Vol. 2176: K.-D. Althoff, R.L. Feldmann, W. Müller (Eds.), Advances in Learning Software Organizations. Proceedings, 2001. XI, 241 pages. 2001.

Vol. 2177: G. Butler, S. Jarzabek (Eds.), Generative and Component-Based Software Engineering. Proceedings, 2001. X, 203 pages. 2001.

Vol. 2178: R. Moreno-Díaz, B. Buchberger, J.-L. Freire (Eds.), Computer Aided Systems Theory – EUROCAST 2001. Proceedings, 2001. XI, 670 pages. 2001.

Vol. 2179: S. Margenov, J. Waśniewski, P. Yalamov (Eds.), Large-Scale Scientific Computing. Proceedings, 2001. XI, 498 pages. 2001.

Vol. 2180: J. Welch (Ed.), Distributed Computing. Proceedings, 2001. X, 343 pages. 2001.

Vol. 2181: C. Y. Westort (Ed.), Digital Earth Moving. Proceedings, 2001. XII, 117 pages. 2001.

Vol. 2182: M. Klusch, F. Zambonelli (Eds.), Cooperative Information Agents V. Proceedings, 2001. XII, 288 pages. 2001. (Subseries LNAI).

Vol. 2183: R. Kahle, P. Schroeder-Heister, R. Stärk (Eds.), Proof Theory in Computer Science. Proceedings, 2001. IX, 239 pages. 2001.

Vol. 2184: M. Tucci (Ed.), Multimedia Databases and Image Communication. Proceedings, 2001. X, 225 pages. 2001.

Vol. 2185: M. Gogolla, C. Kobryn (Eds.), «UML» 2001 – The Unified Modeling Language. Proceedings, 2001. XIV, 510 pages. 2001.

Vol. 2186: J. Bosch (Ed.), Generative and Component-Based Software Engineering. Proceedings, 2001. VIII, 177 pages. 2001.

Vol. 2187: U. Voges (Ed.), Computer Safety, Reliability and Security. Proceedings, 2001. XVI, 249 pages. 2001.

Vol. 2188: F. Bomarius, S. Komi-Sirviö (Eds.), Product Focused Software Process Improvement. Proceedings, 2001. XI, 382 pages. 2001.

Vol. 2189: F. Hoffmann, D.J. Hand, N. Adams, D. Fisher, G. Guimaraes (Eds.), Advances in Intelligent Data Analysis. Proceedings, 2001. XII, 384 pages. 2001.

Vol. 2190: A. de Antonio, R. Aylett, D. Ballin (Eds.), Intelligent Virtual Agents. Proceedings, 2001. VIII, 245 pages. 2001. (Subseries LNAI).

Vol. 2191: B. Radig, S. Florczyk (Eds.), Pattern Recognition. Proceedings, 2001. XVI, 452 pages. 2001.

Vol. 2192: A. Yonezawa, S. Matsuoka (Eds.), Metalevel Architectures and Separation of Crosscutting Concerns. Proceedings, 2001. XI, 283 pages. 2001.

Vol. 2193: F. Casati, D. Georgakopoulos, M.-C. Shan (Eds.), Technologies for E-Services. Proceedings, 2001. X, 213 pages. 2001.

Vol. 2194: A.K. Datta, T. Herman (Eds.), Self-Stabilizing Systems. Proceedings, 2001. VII, 229 pages. 2001.

Vol. 2195: H.-Y. Shum, M. Liao, S.-F. Chang (Eds.), Advances in Multimedia Information Processing – PCM 2001. Proceedings, 2001. XX, 1149 pages. 2001.

Vol. 2196: W. Taha (Ed.), Semantics, Applications, and Implementation of Program Generation. Proceedings, 2001. X, 219 pages. 2001.

Vol. 2197: O. Balet, G. Subsol, P. Torguet (Eds.), Virtual Storytelling. Proceedings, 2001. XI, 213 pages. 2001.

Vol. 2198: N. Zhong, Y. Yao, J. Liu, S. Ohsuga (Eds.), Web Intelligence: Research and Development. Proceedings, 2001. XVI, 615 pages. 2001. (Subseries LNAI).

Vol. 2199: J. Crespo, V. Maojo, F. Martin (Eds.), Medical Data Analysis. Proceedings, 2001. X, 311 pages. 2001.

Vol. 2200: G.I. Davida, Y. Frankel (Eds.), Information Security. Proceedings, 2001. XIII, 554 pages. 2001.

Vol. 2201: G.D. Abowd, B. Brumitt, S. Shafer (Eds.), Ubicomp 2001: Ubiquitous Computing. Proceedings, 2001. XIII, 372 pages. 2001.

Vol. 2202: A. Restivo, S. Ronchi Della Rocca, L. Roversi (Eds.), Theoretical Computer Science. Proceedings, 2001. XI, 440 pages. 2001.

Vol. 2204: A. Brandstädt, V.B. Le (Eds.), Graph-Theoretic Concepts in Computer Science. Proceedings, 2001. X, 329 pages. 2001.

Vol. 2205: D.R. Montello (Ed.), Spatial Information Theory. Proceedings, 2001. XIV, 503 pages. 2001.

Vol. 2206: B. Reusch (Ed.), Computational Intelligence. Proceedings, 2001. XVII, 1003 pages. 2001.

Vol. 2207: I.W. Marshall, S. Nettles, N. Wakamiya (Eds.), Active Networks. Proceedings, 2001. IX, 165 pages. 2001.

Vol. 2208: W.J. Niessen, M.A. Viergever (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001. Proceedings, 2001. XXXV, 1446 pages. 2001.

Vol. 2209: W. Jonker (Ed.), Databases in Telecommunications II. Proceedings, 2001. VII, 179 pages. 2001.

Vol. 2210: Y. Liu, K. Tanaka, M. Iwata, T. Higuchi, M. Yasunaga (Eds.), Evolvable Systems: From Biology to Hardware. Proceedings, 2001. XI, 341 pages. 2001.

Vol. 2211: T.A. Henzinger, C.M. Kirsch (Eds.), Embedded Software. Proceedings, 2001. IX, 504 pages. 2001.

Vol. 2212: W. Lee, L. Mé, A. Wespi (Eds.), Recent Advances in Intrusion Detection. Proceedings, 2001. X, 205 pages. 2001.

Vol. 2213: M.J. van Sinderen, L.J.M. Nieuwenhuis (Eds.), Protocols for Multimedia Systems. Proceedings, 2001. XII, 239 pages. 2001.

Vol. 2214: O. Boldt, H. Jürgensen (Eds.), Automata Implementation. Proceedings, 1999. VIII, 183 pages. 2001.

Vol. 2215: N. Kobayashi, B.C. Pierce (Eds.), Theoretical Aspects of Computer Software. Proceedings, 2001. XV, 561 pages. 2001.

Vol. 2216: E.S. Al-Shaer, G. Pacifici (Eds.), Management of Multimedia on the Internet. Proceedings, 2001. XIV, 373 pages. 2001.

Vol. 2217: T. Gomi (Ed.), Evolutionary Robotics. Proceedings, 2001. XI, 139 pages. 2001.

Vol. 2218: R. Guerraoui (Ed.), Middleware 2001. Proceedings, 2001. XIII, 395 pages. 2001.

Vol. 2219: S.T. Taft, R.A. Duff, R.L. Brukardt, E. Ploedereder (Eds.), Consolidated Ada Reference Manual. XXV, 560 pages. 2001.

Vol. 2220: C. Johnson (Ed.), Interactive Systems. Proceedings, 2001. XII, 219 pages. 2001.

Vol. 2221: D.G. Feitelson, L. Rudolph (Eds.), Job Scheduling Strategies for Parallel Processing. Proceedings, 2001. VII, 207 pages. 2001.

Vol. 2223: P. Eades, T. Takaoka (Eds.), Algorithms and Computation. Proceedings, 2001. XIV, 780 pages. 2001.

Vol. 2224: H.S. Kunii, S. Jajodia, A. Sølvberg (Eds.), Conceptual Modeling – ER 2001. Proceedings, 2001. XIX, 614 pages. 2001.

Vol. 2225: N. Abe, R. Khardon, T. Zeugmann (Eds.), Algorithmic Learning Theory. Proceedings, 2001. XI, 379 pages. 2001. (Subseries LNAI).

Vol. 2226: K.P. Jantke, A. Shinohara (Eds.), Discovery Science. Proceedings, 2001. XII, 494 pages. 2001. (Subseries LNAI).

Vol. 2227: S. Boztaş, I.E. Shparlinski (Eds.), Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. Proceedings, 2001. XII, 398 pages. 2001.

Vol. 2228: B. Monien, V.K. Prasanna, S. Vajapeyam (Eds.), High Performance Computing – HiPC 2001. Proceedings, 2001. XVIII, 438 pages. 2001.

Vol. 2229: S. Qing, T. Okamoto, J. Zhou (Eds.), Information and Communications Security. Proceedings, 2001. XIV, 504 pages. 2001.

Vol. 2230: T. Katila, I.E. Magnin, P. Clarysse, J. Montagnat, J. Nenonen (Eds.), Functional Imaging and Modeling of the Heart. Proceedings, 2001. XI, 158 pages. 2001.

Vol. 2232: L. Fiege, G. Mühl, U. Wilhelm (Eds.), Electronic Commerce. Proceedings, 2001. X, 233 pages. 2001.

Vol. 2233: J. Crowcroft, M. Hofmann (Eds.), Networked Group Communication. Proceedings, 2001. X, 205 pages. 2001.

Vol. 2234: L. Pacholski, P. Ružička (Eds.), SOFSEM 2001: Theory and Practice of Informatics. Proceedings, 2001. XI, 347 pages. 2001.

Vol. 2235: C.S. Calude, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Multiset Processing. VIII, 359 pages. 2001.

Vol. 2237: P. Codognet (Ed.), Logic Programming. Proceedings, 2001. XI, 365 pages. 2001.

Vol. 2239: T. Walsh (Ed.), Principles and Practice of Constraint Programming – CP 2001. Proceedings, 2001. XIV, 788 pages. 2001.

Vol. 2240: G.P. Picco (Ed.), Mobile Agents. Proceedings, 2001. XIII, 277 pages. 2001.

Vol. 2241: M. Jünger, D. Naddef (Eds.), Computational Combinatorial Optimization. IX, 305 pages. 2001.

Vol. 2242: C.A. Lee (Ed.), Grid Computing – GRID 2001. Proceedings, 2001. XII, 185 pages. 2001.

Vol. 2244: D. Bjørner, M. Broy, A.V. Zamulin (Eds.), Perspectives of System Informatics. Proceedings, 2001. XIII, 548 pages. 2001.

Vol. 2245: R. Hariharan, M. Mukund, V. Vinay (Eds.), FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science. Proceedings, 2001. XI, 347 pages. 2001.

Vol. 2246: R. Falcone, M. Singh, Y.-H. Tan (Eds.), Trust in Cyber-societies. VIII, 195 pages. 2001. (Subseries LNAI).

Vol. 2247: C. P. Rangan, C. Ding (Eds.), Progress in Cryptology – INDOCRYPT 2001. Proceedings, 2001. XIII, 351 pages. 2001.

Vol. 2248: C. Boyd (Ed.), Advances in Cryptology – ASIACRYPT 2001. Proceedings, 2001. XI, 603 pages. 2001.

Vol. 2249: K. Nagi, Transactional Agents. XVI, 205 pages. 2001.

Vol. 2250: R. Nieuwenhuis, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. Proceedings, 2001. XV, 738 pages. 2001. (Subseries LNAI).

Vol. 2251: Y.Y. Tang, V. Wickerhauser, P.C. Yuen, C.Li (Eds.), Wavelet Analysis and Its Applications. Proceedings, 2001. XIII, 450 pages. 2001.

Vol. 2252: J. Liu, P.C. Yuen, C. Li, J. Ng, T. Ishida (Eds.), Active Media Technology. Proceedings, 2001. XII, 402 pages. 2001.

Vol. 2253: T. Terano, T. Nishida, A. Namatame, S. Tsumoto, Y. Ohsawa, T. Washio (Eds.), New Frontiers in Artificial Intelligence. Proceedings, 2001. XXVII, 553 pages. 2001. (Subseries LNAI).

Vol. 2254: M.R. Little, L. Nigay (Eds.), Engineering for Human-Computer Interaction. Proceedings, 2001. XI, 359 pages. 2001.

Vol. 2256: M. Stumptner, D. Corbett, M. Brooks (Eds.), AI 2001: Advances in Artificial Intelligence. Proceedings, 2001. XII, 666 pages. 2001. (Subseries LNAI).

Vol. 2258: P. Brazdil, A. Jorge (Eds.), Progress in Artificial Intelligence. Proceedings, 2001. XII, 418 pages. 2001. (Subseries LNAI).

Vol. 2259: S. Vaudenay, A.M. Youssef (Eds.), Selected Areas in Cryptography. Proceedings, 2001. XI, 359 pages. 2001.

Vol. 2260: B. Honary (Ed.), Cryptography and Coding. Proceedings, 2001. IX, 416 pages. 2001.

Vol. 2264: K. Steinhöfel (Ed.), Stochastic Algorithms: Foundations and Applications. Proceedings, 2001. VIII, 203 pages. 2001.

Preface

SAC 2001, the eighth annual workshop on selected areas in cryptography, was held at the Fields Institute in Toronto, Ontario, Canada. Previous SAC workshops were held at Queen's University in Kingston (1994, 1996, 1998, and 1999), at Carlton University in Ottawa (1995 and 1997) and at the University of Waterloo (2000). The conference was sponsored by the center for applied cryptographic research (CACR) at the University of Waterloo, Certicom Corporation, Communications and Information Technology Ontario (CITO), Ecole Polytechnique Fédérale de Lausanne, Entrust Technologies, and ZeroKnowledge. We are grateful to these organizations for their support of the conference.

The current SAC board includes Carlisle Adams, Doug Stinson, Ed Dawson, Henk Meijer, Howard Heys, Michael Wiener, Serge Vaudenay, Stafford Tavares, and Tom Cusick. We would like to thank all of them for giving us the mandate to organize SAC 2001.

The themes for SAC 2001 workshop were:

- Design and analysis of symmetric key cryptosystems.
- Primitives for private key cryptography, including block and stream ciphers, hash functions, and MACs.
- Efficient implementations of cryptographic systems in public and private key cryptography.
- Cryptographic solutions for web and internet security.

There were 57 technical papers submitted to the conference from an international authorship. Every paper was refereed by at least 3 reviewers and 25 papers were accepted for presentation at the conference. We would like to thank the authors of all the submitted papers, both those whose work is included in these proceedings, and those whose work could not be accommodated.

In addition to these 25 papers, two invited presentations were given at the conference: one by Moti Yung from CertCo, USA, entitled "Polynomial Reconstruction Based Cryptography " and the other by Phong Nguyen from the Ecole Normale Supérieure, France, entitled "The two faces of lattices in cryptology". Thanks to both Moti and Phong for their excellent talks and for kindly accepting our invitation.

The program committee for SAC 2001 consisted of the following members: Stefan Brands, Matt Franklin, Henri Gilbert, Howard Heys, Hideki Imai, Shiho Moriai, Kaisa Nyberg, Rich Schroeppel, Doug Stinson, Stafford Tavares, Serge Vaudenay, Michael Wiener, Amr Youssef , and Yuliang Zheng.

On behalf of the program committee we would like to thank the following sub-referees for their help in the reviewing process: Joonsang Baek, Guang Gong, Ian Goldberg, Darrel Hankerson, Keiichi Iwamura, Mike Just, Masayuki Kanda, Liam Keliher, Mira Kim, Kazukuni Kobara, Frédéric Légaré, Henk Meijer, Alfred John Menezes, Miodrag Mihaljevic, Ulf Mőller, Dalit Naor, Daisuke Nojiri, Mohammad Ghulam Rahman, Palash Sarkar, Akashi Satoh, Junji Shikata, Takeshi Shimoyama, Ron Steinfeld, Anton Stiglic, Edlyn Teske, Yodai Watanabe, Huapeng Wu, Daichi Yamane, and Robert Zuccherato.

We would like to thank all the people involved in organizing the conference. In particular we would like to thank Pascal Junod for his effort in making the reviewing process run smoothly. Special thanks are due to Frances Hannigan for her help in the local arrangements and for making sure that everything ran smoothly during the workshop. Finally we would like to thank all the participants of SAC 2001.

August 2001 Serge Vaudenay and Amr Youssef

Organization

Program Committee

S. Brands	Zero Knowledge Systems (Canada)
M. Franklin	UC Davis (USA)
H. Gilbert	France Telecom (France)
H. Heys	Memorial University of Newfoundland (Canada)
H. Imai	University of Tokyo (Japan)
S. Moriai	NTT (Japan)
K. Nyberg	Nokia Research Center (Finland)
R. Schroeppel	Sandia National Lab (USA)
D. Stinson	University of Waterloo (Canada)
S. Tavares	Queen's University (Canada)
S. Vaudenay (co-chair)	EPFL (Switzerland)
M. Wiener	Entrust Technologies (Canada)
A. Youssef (co-chair)	University of Waterloo (Canada)
Y. Zheng	Monash University (Australia)

Local Organizing Committee

S. Vaudenay	EPFL (Switzerland)
A. Youssef	University of Waterloo (Canada)
F. Hannigan	University of Waterloo (Canada)
P. Junod	EPFL (Switzerland)

Sponsoring Institutions

EPFL
University of Waterloo
Entrust Technologies
Certicom
Zero Knowledge Systems
CITO

Table of Contents

Invited Talk I

Elliptic Curves and Efficient Implementation I

Cryptanalysis II

Elliptic Curves and Efficient Implementation II

Public Key Systems

Invited Talk II

Protocols and Mac

Weaknesses in the Key Scheduling Algorithm of RC4

Scott Fluhrer[1], Itsik Mantin[2], and Adi Shamir[2]

[1] Cisco Systems, Inc.,
170 West Tasman Drive, San Jose, CA 95134, USA
sfluhrer@cisco.com
[2] Computer Science department, The Weizmann Institute,
Rehovot 76100, Israel
{itsik,shamir}@wisdom.weizmann.ac.il

Abstract. In this paper we present several weaknesses in the key scheduling algorithm of RC4, and describe their cryptanalytic significance. We identify a large number of weak keys, in which knowledge of a small number of key bits suffices to determine many state and output bits with non-negligible probability. We use these weak keys to construct new distinguishers for RC4, and to mount related key attacks with practical complexities. Finally, we show that RC4 is completely insecure in a common mode of operation which is used in the widely deployed Wired Equivalent Privacy protocol (WEP, which is part of the 802.11 standard), in which a fixed secret key is concatenated with known IV modifiers in order to encrypt different messages. Our new passive ciphertext-only attack on this mode can recover an arbitrarily long key in a negligible amount of time which grows only linearly with its size, both for 24 and 128 bit IV modifiers.

1 Introduction

RC4 is the most widely used stream cipher in software applications. It was designed by Ron Rivest in 1987 and kept as a trade secret until it leaked out in 1994. RC4 has a secret internal state which is a permutation of all the $N = 2^n$ possible n bits words, along with two indices in it. In practical applications $n = 8$, and thus RC4 has a huge state of $log_2(2^8! \times (2^8)^2) \approx 1700$ bits.

In this paper we analyze the Key Scheduling Algorithm (KSA) which derives the initial state from a variable size key, and describe two significant weaknesses of this process. The first weakness is the existence of large classes of weak keys, in which a small part of the secret key determines a large number of bits of the initial permutation (KSA output). In addition, the Pseudo Random Generation Algorithm (PRGA) translates these patterns in the initial permutation into patterns in the prefix of the output stream, and thus RC4 has the undesirable property that for these weak keys its initial outputs are disproportionally affected by a small number of key bits. These weak keys have length which is

S. Vaudenay and A. Youssef (Eds.): SAC 2001, LNCS 2259, pp. 1–24, 2001.

divisible by some non-trivial power of two, i.e., $\ell = 2^q m$ for some $q > 0$[1]. When $RC4_n$ uses such a weak key of ℓ words, fixing $n + q(\ell - 1) + 1$ bits of K (as a particular pattern) determines $\Theta(qN)$ bits of the initial permutation with probability of one half and determines various prefixes of the output stream with various probabilities (depending on their length).

The second weakness is a related key vulnerability, which applies when part of the key presented to the KSA is exposed to the attacker. It consists of the observation that when the same secret part of the key is used with numerous different exposed values, an attacker can rederive the secret part by analyzing the initial word of the keystreams with relatively little work. This concatenation of a long term secret part with an attacker visible part is a commonly used mode of RC4, and in particular it is used in the WEP (Wired Equivalent Privacy) protocol, which protects many wireless networks. Our new attack on this mode is practical for any key size and for any modifier size, including the 24 bit recommended in the original WEP, and the 128 bit recommended in the revised version WEP2.

The paper is organized in the following way: In Section 2 we describe RC4 and previous results about its security. In Section 3 we consider a slightly modified variant of the Key Scheduling Algorithm, called KSA*, and prove that a particular pattern of a small number of key bits suffices to completely determine a large number of state bits. Afterwards, we show that this weakness of KSA*, which we denote as the *invariance weakness*, exists (in a weaker form) also in the original KSA. In Section 4 we show that with high probability, the patterns of initial states associated with these weak keys also propagate into the first few outputs, and thus a small number of weak key bits determine a large number of bits in the output stream. In Section 5 we describe several cryptanalytic applications of the invariance weakness, including a new type of distinguisher. In Sections 6 and 7 we describe the second weakness, which we denote as the *IV weakness*, and show that a common method of using RC4 is vulnerable to a practical attack due to this weakness. In Section 8, we show how both these weaknesses can separately be used in a related key attack. In the appendices, we examine how the IV weakness can be used to attack a real system (appendix A), how the invariance weakness can be used to construct a ciphertext-only distinguisher and to prove that RC4 has low sampling resistance (appendices B and C), and how to derive the secret key from an early permutation state (appendix D).

2 RC4 and Its Security

2.1 Description of RC4

RC4 consists of two parts (described in Figure 1): A key scheduling algorithm KSA which turns a random key (whose typical size is 40-256 bits) into an initial

[1] Here and in the rest of the paper ℓ is the number of words of K, where each word contains n bits.

KSA(K)
Initialization:
For $i = 0 \ldots N-1$
$S[i] = i$
$j = 0$
Scrambling:
For $i = 0 \ldots N-1$
$j = j + S[i] + K[i \bmod \ell]$
$Swap(S[i], S[j])$

PRGA(K)
Initialization:
$i = 0$
$j = 0$
Generation loop:
$i = i + 1$
$j = j + S[i]$
Swap$(S[i], S[j])$
Output $z = S[S[i] + S[j]]$

Fig. 1. The Key Scheduling Algorithm and the Pseudo-Random Generation Algorithm

permutation S of $\{0, \ldots, N-1\}$, and an output generation part PRGA which uses this permutation to generate a pseudo-random output sequence.

The PRGA initializes two indices i and j to 0, and then loops over four simple operations which increment i as a counter, increment j pseudo randomly, exchange the two values of S pointed to by i and j, and output the value of S pointed to by $S[i] + S[j]$[2]. Note that every entry of S is swapped at least once (possibly with itself) within any N consecutive rounds, and thus the permutation S evolves fairly rapidly during the output generation process.

The KSA consists of N loops that are similar to the PRGA round operation. It initializes S to be the identity permutation and i and j to 0, and applies the PRGA round operation N times, stepping i across S, and updating j by adding $S[i]$ and the next word of the key (in cyclic order).

2.2 Previous Attacks on RC4

Due to the huge effective key of RC4, attacking the PRGA seems to be infeasible (the best known attack on this part requires time that exceeds 2^{700}). The only practical results related to the PRGA deal with the construction of distinguishers. Fluhrer and McGrew described in [FM00] how to distinguish RC4 outputs from random strings with 2^{30} data. A better distinguisher which requires 2^8 data was described by Mantin and Shamir in [MS01]. However, this distinguisher could only be used to mount a partial attack on RC4 in broadcast applications.

The fact that the initialization of RC4 is very simple stimulated considerable research on this mechanism of RC4. In particular, Roos discovered in [Roo95] a class of weak keys that reduces their effective size by five bits, and Grosul and Wallach showed in [GW00] that for large keys whose size is close to N words, RC4 is vulnerable to a related key attack.

[2] Here and in the rest of the paper all the additions are carried out modulo N

More analysis of the security of RC4 can be found in [KMP+98], [Gol97] and [MT98].

3 The Invariance Weakness

Due to space limitations we prove here the invariance weakness only for a simplified variant of the KSA, which we denote as KSA* and describe in Figure 2. The only difference between them is that KSA* updates i at the *beginning* of the loop, whereas KSA updates i at the *end* of the loop. After formulating and proving the existence of this weakness in KSA*, we describe the modifications required to apply this analysis to the real KSA.

3.1 Definitions

We start the round numbering from 0, which means that both KSA and KSA* have rounds $0, \ldots, N-1$. We denote the indices swapped in round r by i_r and j_r, and the permutation S after swapping these indices is denoted as S_r. Notice that by using this notation, $i_r = r$ in the real KSA. However, in KSA* this notation becomes somewhat confusing, when $i_r = r+1$. For the sake of completeness, we can say that $j_{-1} = 0$, S_{-1} is the identity permutation and $i_{-1} = \begin{cases} -1 & KSA \\ 0 & KSA^* \end{cases}$.

Definition 1. *Let S be a permutation of $\{0, \ldots, N-1\}$, t be an index in S and b be some integer. Then if $S[t] \overset{\text{mod } b}{\equiv} t$, the permutation S is said to* b-conserve *the index t. Otherwise, the permutation S is said to* b-unconserve *the index t.*

Definition 2. *A permutation S of $\{0, \ldots, N-1\}$ is* b-conserving *if $I_b(S) = N$, and is* almost b-conserving *if $I_b(S) \geq N-2$.*

KSA(K)[a]	KSA*(K)
For $i = 0 \ldots N-1$	For $i = 0 \ldots N-1$
$S[i] = i$	$S[i] = i$
$i = 0$	$i = 0$
$j = 0$	$j = 0$
Repeat N times	Repeat N times
$j = j + S[i] + K[i \bmod \ell]$	$i = i + 1$
$Swap(S[i], S[j])$	$j = j + S[i] + K[i \bmod \ell]$
$i = i + 1$	$Swap(S[i], S[j])$

[a] KSA is rewritten in a way which clarifies its relation to KSA*

Fig. 2. KSA vs. KSA*

We denote the number of indices that a permutation b-conserves as $I_b(S)$. To simplify the notation, we often write I_r instead of $I_b(S_r)$.

Definition 3. *Let b, ℓ be integers, and let K be an ℓ word key. Then K is called a* b-exact *key if for any index r, $K[r \bmod \ell] \equiv (1-r) \pmod b$. In case $K[0] = 1$ and $MSB(K[1]) = 1$, K is called a* special b-exact *key.*

Notice that for this condition to hold, it is necessary (but not sufficient) that $b \mid \ell$.

3.2 The Weakness

Theorem 1. *Let $q \leq n$ and ℓ be integers and $b \stackrel{def}{=} 2^q$. Suppose that $b \mid \ell$ and let K be a b-exact key of ℓ words. Then the permutation $S = KSA^*(K)$ is b-conserving.*

Before getting to the proof itself, we will prove an auxiliary lemma

Lemma 1. *If $i_{r+1} \equiv j_{r+1} \pmod b$, then $I_{r+1} = I_r$.*

Proof. The only operation that might affect S (and maybe I) is the swapping operation. However, when i and j are equivalent (mod b) in round $r+1$, S_{r+1} b-conserves position i_{r+1} (j_{r+1}) if and only if S_r b-conserved position j_r (i_r). Thus the number of indices S b-conserves remains the same.

Proof. (of Theorem 1) We will prove by induction on r that for any $-1 \leq r \leq N-1$, it turns out that $i_r \equiv j_r \pmod b$ and $I_b(S_r) = N$ and . This in particular implies that $I_{N-1} = N$, which makes the output permutation b-conserving.

For $r = -1$ (before the first round), the claim is trivial because $i_{-1} = j_{-1} = 0$ and S_{-1} is the identity permutation which is b-conserving for every b. Suppose that $j_r \equiv i_r$ and S_r is b-conserving. Then $i_{r+1} = i_r + 1$ and

$$j_{r+1} = j_r + S_r[i_{r+1}] + K[i_{r+1} \bmod \ell] \stackrel{\bmod b}{\equiv} i_r + i_{r+1} + (1 - i_{r+1}) = i_r + 1 = i_{r+1}$$

Thus, $i_{r+1} \equiv j_{r+1} \pmod b$ and by applying Lemma 1 we get $I_{r+1} = I_r = N$ and therefore S_{r+1} is b-conserving.

KSA* thus transforms special patterns in the key into corresponding patterns in the initial permutation. The fraction of determined permutation bits is proportional to the fraction of fixed key bits. For example, applying this result to $RC4_{n=8,\ell=6}$ and $q = 1$, 6 out of the 48 key bits completely determine 252 out of the 1684 permutation bits (this is the number of bits encapsulated in the LSBs).

3.3 Adjustments to KSA

The small difference between KSA* and KSA (see Figure 2) is essential in that KSA, applied to a b-exact key, does not preserve the equivalence (mod b) of i and j even after the first round. Analyzing its execution on a b-exact key gives

$$j_0 = j_{-1} + S_{-1}[i_0] + K[i_0] = 0 + S_{-1}[0] + K[0] = K[0] \stackrel{\bmod b}{\equiv} 1 \stackrel{\bmod b}{\not\equiv} 0 = i_0$$

and thus the structure described in Section 3.2 cannot be preserved by the cyclic use of the key words. However, it is possible to adjust the invariance weakness to the real KSA, and the proper modifications are formulated in the following theorem:

Theorem 2. *Let $q \leq n$ and ℓ be integers and $b \stackrel{def}{=} 2^q$. Suppose that $b \mid \ell$ and let K be a special b-exact key of ℓ words. Then*

$$Pr[KSA(K) \text{ is almost } b\text{-conserving}] \geq 2/5$$

where the probability is over the rest of the key bits.

Due to space limitations, the formal proof of this theorem (which is based on a detailed case analysis) will appear only in the full version of this paper. However, we can explain the intuition behind this theorem by concentrating on the differences between Theorems 1 and 2, which deal with KSA* and KSA respectively. During the first round, two deviations from KSA* execution occur. The first one is the non-equivalence of i and j which is expected to cause non-equivalent entries to be swapped during the next rounds, thus ruining the delicate structure that was preserved so well during KSA* execution. The second deviation is that S b-unconserves two of the indices, $i_0 = 0$ and $j_0 = K[0]$. However, we can cancel the ij discrepancy by forcing $K[0]$ (and j_0) to 1. In this case, the discrepancy in $S[j_0]$ ($S[1]$) causes an improper value to be added to j in round 1, thus repairing its non-equivalence to i during this round. At this point there are still two unconserved indices, and this aberration is dragged across the whole execution into the resulting permutation. Although these corrupted entries might interfere with j updates, the pseudo-random j might reach them *before* they are used to update j (i.e., before i reaches them), and send them into a region in S where they cannot affect the next values of j[3]. The probability of this lucky event is amplified by the fact that the corrupted entries are $i_0 = 0$ which is not touched (by i) until the termination of the KSA due to its distance from the current location of i, and $j_1 = 1 + K[1] > N/2$ (recall that $MSB(K[1]) = 1$), that is far the position of i ($i_1 = 1$), which gives j many opportunities to reach it before i does. The probability of $N/2$ pseudo random j's to reach an arbitrary value can be bounded from below by 2/5, and extensive experimentation indicates that this probability is actually close to one half.

4 Key-Output Correlation

In this section we will analyze the propagation of the weak key patterns into the generated outputs. First we prove Claim 4 which deals with the highly biased behavior of a significantly weakened variant of the PRGA (where the swaps are avoided), applied to a b-conserving permutation. Next, we will argue that the

[3] if a value is pointed to by j before the swap, it will not be used as $S[i]$ (before the swap) for at least $N - 1$ rounds, and in particular it will not affect the values of j during these rounds.