Lecture Notes in
Computer Science 1758

Howard Heys Carlisle Adams (Eds.)

Selected Areas in Cryptography

6th Annual International Workshop, SAC'99
Kingston, Ontario, Canada, August 1999
Proceedings

Springer

Howard Heys Carlisle Adams (Eds.)

Selected Areas in Cryptography

6th Annual International Workshop, SAC'99
Kingston, Ontario, Canada, August 9-10, 1999
Proceedings

Springer

cryptography

Volume Editors

Howard Heys
Faculty of Engineering and Applied Science
Memorial University of Newfoundland
St. John's, Newfoundland, Canada A1B 3X5
E-mail: howard@engr.mun.ca

Carlisle Adams
Entrust Technologies
750 Heron Road, Suite E08
Ottawa, Ontario, Canada K1V 1A7
E-mail: cadams@entrust.com

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Selected areas in cryptography : 6th annual international workshop ;
proceedings / SAC '99, Kingston, Ontario, Canada, August 9 - 11,
1999. Howard Heys ; Carlisle Adams (ed.). - Berlin ; Heidelberg ; New
York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ;
Tokyo : Springer, 2000
(Lecture notes in computer science ; Vol. 1758)
ISBN 3-540-67185-4

CR Subject Classification (1991): E.3, C.2, D.4.6, K.6.5, F.2.1-2, H.4.3

ISSN 0302-9743
ISBN 3-540-67185-4 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag is a company in the specialist publishing group BertelsmannSpringer
© Springer-Verlag Berlin Heidelberg 2000
Printed in Germany

Typesetting: Camera-ready by author
Printed on acid-free paper SPIN 10719627 57/3144 5 4 3 2 1 0

Lecture Notes in Computer Science 1758

Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Lecture Notes in Computer Science

For information about Vols. 1–1693
please contact your bookseller or Springer-Verlag

Vol. 1694: A. Cortesi, G. Filé (Eds.), Static Analysis. Proceedings, 1999. VIII, 357 pages. 1999.

Vol. 1695: P. Barahona, J.J. Alferes (Eds.), Progress in Artificial Intelligence. Proceedings, 1999. XI, 385 pages. 1999. (Subseries LNAI).

Vol. 1696: S. Abiteboul, A.-M. Vercoustre (Eds.), Research and Advanced Technology for Digital Libraries. Proceedings, 1999. XII, 497 pages. 1999.

Vol. 1697: J. Dongarra, E. Luque, T. Margalef (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. Proceedings, 1999. XVII, 551 pages. 1999.

Vol. 1698: M. Felici, K. Kanoun, A. Pasquini (Eds.), Computer Safety, Reliability and Security. Proceedings, 1999. XVIII, 482 pages. 1999.

Vol. 1699: S. Albayrak (Ed.), Intelligent Agents for Telecommunication Applications. Proceedings, 1999. IX, 191 pages. 1999. (Subseries LNAI).

Vol. 1700: R. Stadler, B. Stiller (Eds.), Active Technologies for Network and Service Management. Proceedings, 1999. XII, 299 pages. 1999.

Vol. 1701: W. Burgard, T. Christaller, A.B. Cremers (Eds.), KI-99: Advances in Artificial Intelligence. Proceedings, 1999. XI, 311 pages. 1999. (Subseries LNAI).

Vol. 1702: G. Nadathur (Ed.), Principles and Practice of Declarative Programming. Proceedings, 1999. X, 434 pages. 1999.

Vol. 1703: L. Pierre, T. Kropf (Eds.), Correct Hardware Design and Verification Methods. Proceedings, 1999. XI, 366 pages. 1999.

Vol. 1704: Jan M. Żytkow, J. Rauch (Eds.), Principles of Data Mining and Knowledge Discovery. Proceedings, 1999. XIV, 593 pages. 1999. (Subseries LNAI).

Vol. 1705: H. Ganzinger, D. McAllester, A. Voronkov (Eds.), Logic for Programming and Automated Reasoning. Proceedings, 1999. XII, 397 pages. 1999. (Subseries LNAI).

Vol. 1706: J. Hatcliff, T. Æ. Mogensen, P. Thiemann (Eds.), Partial Evaluation – Practice and Theory. 1998. IX, 433 pages. 1999.

Vol. 1707: H.-W. Gellersen (Ed.), Handheld and Ubiquitous Computing. Proceedings, 1999. XII, 390 pages. 1999.

Vol. 1708: J.M. Wing, J. Woodcock, J. Davies (Eds.), FM'99 – Formal Methods. Proceedings Vol. I, 1999. XVIII, 937 pages. 1999.

Vol. 1709: J.M. Wing, J. Woodcock, J. Davies (Eds.), FM'99 – Formal Methods. Proceedings Vol. II, 1999. XVIII, 937 pages. 1999.

Vol. 1710: E.-R. Olderog, B. Steffen (Eds.), Correct System Design. XIV, 417 pages. 1999.

Vol. 1711: N. Zhong, A. Skowron, S. Ohsuga (Eds.), New Directions in Rough Sets, Data Mining, and Granular-Soft Computing. Proceedings, 1999. XIV, 558 pages. 1999. (Subseries LNAI).

Vol. 1712: H. Boley, A Tight, Practical Integration of Relations and Functions. XI, 169 pages. 1999. (Subseries LNAI).

Vol. 1713: J. Jaffar (Ed.), Principles and Practice of Constraint Programming – CP'99. Proceedings, 1999. XII, 493 pages. 1999.

Vol. 1714: M.T. Pazienza (Eds.), Information Extraction. IX, 165 pages. 1999. (Subseries LNAI).

Vol. 1715: P. Perner, M. Petrou (Eds.), Machine Learning and Data Mining in Pattern Recognition. Proceedings, 1999. VIII, 217 pages. 1999. (Subseries LNAI).

Vol. 1716: K.Y. Lam, E. Okamoto, C. Xing (Eds.), Advances in Cryptology – ASIACRYPT'99. Proceedings, 1999. XI, 414 pages. 1999.

Vol. 1717: Ç. K. Koç, C. Paar (Eds.), Cryptographic Hardware and Embedded Systems. Proceedings, 1999. XI, 353 pages. 1999.

Vol. 1718: M. Diaz, P. Owezarski, P. Sénac (Eds.), Interactive Distributed Multimedia Systems and Telecommunication Services. Proceedings, 1999. XI, 386 pages. 1999.

Vol. 1719: M. Fossorier, H. Imai, S. Lin, A. Poli (Eds.), Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. Proceedings, 1999. XIII, 510 pages. 1999.

Vol. 1720: O. Watanabe, T. Yokomori (Eds.), Algorithmic Learning Theory. Proceedings, 1999. XI, 365 pages. 1999. (Subseries LNAI).

Vol. 1721: S. Arikawa, K. Furukawa (Eds.), Discovery Science. Proceedings, 1999. XI, 374 pages. 1999. (Subseries LNAI).

Vol. 1722: A. Middeldorp, T. Sato (Eds.), Functional and Logic Programming. Proceedings, 1999. X, 369 pages. 1999.

Vol. 1723: R. France, B. Rumpe (Eds.), UML'99 – The Unified Modeling Language. XVII, 724 pages. 1999.

Vol. 1724: H. I. Christensen, H. Bunke, H. Noltemeier (Eds.), Sensor Based Intelligent Robots. Proceedings, 1998. VIII, 327 pages. 1999 (Subseries LNAI).

Vol. 1725: J. Pavelka, G. Tel, M. Bartošek (Eds.), SOFSEM'99: Theory and Practice of Informatics. Proceedings, 1999. XIII, 498 pages. 1999.

Vol. 1726: V. Varadharajan, Y. Mu (Eds.), Information and Communication Security. Proceedings, 1999. XI, 325 pages. 1999.

Vol. 1727: P.P. Chen, D.W. Embley, J. Kouloumdjian, S.W. Liddle, J.F. Roddick (Eds.), Advances in Conceptual Modeling. Proceedings, 1999. XI, 389 pages. 1999.

Vol. 1728: J. Akoka, M. Bouzeghoub, I. Comyn-Wattiau, E. Métais (Eds.), Conceptual Modeling – ER '99. Proceedings, 1999. XIV, 540 pages. 1999.

Vol. 1729: M. Mambo, Y. Zheng (Eds.), Information Security. Proceedings, 1999. IX, 277 pages. 1999.

Vol. 1730: M. Gelfond, N. Leone, G. Pfeifer (Eds.), Logic Programming and Nonmonotonic Reasoning. Proceedings, 1999. XI, 391 pages. 1999. (Subseries LNAI).

Vol. 1731: J. Kratochvíl (Ed.), Graph Drawing. Proceedings, 1999. XIII, 422 pages. 1999.

Vol. 1732: S. Matsuoka, R.R. Oldehoeft, M. Tholburn (Eds.), Computing in Object-Oriented Parallel Environments. Proceedings, 1999. VIII, 205 pages. 1999.

Vol. 1733: H. Nakashima, C. Zhang (Eds.), Approaches to Intelligent Agents. Proceedings, 1999. XII, 241 pages. 1999. (Subseries LNAI).

Vol. 1734: H. Hellwagner, A. Reinefeld (Eds.), SCI: Scalable Coherent Interface. XXI, 490 pages. 1999.

Vol. 1564: M. Vazirgiannis, Interactive Multimedia Documents. XIII, 161 pages. 1999.

Vol. 1591: D.J. Duke, I. Herman, M.S. Marshall, PREMO: A Framework for Multimedia Middleware. XII, 254 pages. 1999.

Vol. 1624: J. A. Padget (Ed.), Collaboration between Human and Artificial Societies. XIV, 301 pages. 1999. (Subseries LNAI).

Vol. 1635: X. Tu, Artificial Animals for Computer Animation. XIV, 172 pages. 1999.

Vol. 1646: B. Westfechtel, Models and Tools for Managing Development Processes. XIV, 418 pages. 1999.

Vol. 1735: J.W. Amtrup, Incremental Speech Translation. XV, 200 pages. 1999. (Subseries LNAI).

Vol. 1736: L. Rizzo, S. Fdida (Eds.): Networked Group Communication. Proceedings, 1999. XIII, 339 pages. 1999.

Vol. 1737: P. Agouris, A. Stefanidis (Eds.), Integrated Spatial Databases. Proceedings, 1999. X, 317 pages. 1999.

Vol. 1738: C. Pandu Rangan, V. Raman, R. Ramanujam (Eds.), Foundations of Software Technology and Theoretical Computer Science. Proceedings, 1999. XII, 452 pages. 1999.

Vol. 1739: A. Braffort, R. Gherbi, S. Gibet, J. Richardson, D. Teil (Eds.), Gesture-Based Communication in Human-Computer Interaction. Proceedings, 1999. XI, 333 pages. 1999. (Subseries LNAI).

Vol. 1740: R. Baumgart (Ed.): Secure Networking – CQRE [Secure] '99. Proceedings, 1999. IX, 261 pages. 1999.

Vol. 1741: A. Aggarwal, C. Pandu Rangan (Eds.), Algorithms and Computation. Proceedings, 1999. XIII, 448 pages. 1999.

Vol. 1742: P.S. Thiagarajan, R. Yap (Eds.), Advances in Computing Science – ASIAN'99. Proceedings, 1999. XI, 397 pages. 1999.

Vol. 1743: A. Moreira, S. Demeyer (Eds.), Object-Oriented Technology. Proceedings, 1999. XVII, 389 pages. 1999.

Vol. 1744: S. Staab, Extracting Degree Information from Texts. X; 187 pages. 1999. (Subseries LNAI).

Vol. 1745: P. Banerjee, V.K. Prasanna, B.P. Sinha (Eds.), High Performance Computing – HiPC'99. Proceedings, 1999. XXII, 412 pages. 1999.

Vol. 1746: M. Walker (Ed.), Cryptography and Coding. Proceedings, 1999. IX, 313 pages. 1999.

Vol. 1747: N. Foo (Ed.), Adavanced Topics in Artificial Intelligence. Proceedings, 1999. XV, 500 pages. 1999. (Subseries LNAI).

Vol. 1748: H.V. Leong, W.-C. Lee, B. Li, L. Yin (Eds.), Mobile Data Access. Proceedings, 1999. X, 245 pages. 1999.

Vol. 1749: L. C.-K. Hui, D.L. Lee (Eds.), Internet Applications. Proceedings, 1999. XX, 518 pages. 1999.

Vol. 1750: D.E. Knuth, MMIXware. VIII, 550 pages. 1999.

Vol. 1751: H. Imai, Y. Zheng (Eds.), Public Key Cryptography. Proceedings, 2000. XI, 485 pages. 2000.

Vol. 1753: E. Pontelli, V. Santos Costa (Eds.), Practical Aspects of Declarative Languages. Proceedings, 2000. X, 327 pages. 2000.

Vol. 1754: J. Väänänen (Ed.), Generalized Quantifiers and Computation. Proceedings, 1997. VII, 139 pages. 1999.

Vol. 1755: D. Bjørner, M. Broy, A.V. Zamulin (Eds.), Perspectives of System Informatics. Proceedings, 1999. XII, 540 pages. 2000.

Vol. 1758: H. Heys, C. Adams (Eds.), Selected Areas in Cryptography. Proceedings, 1999. VIII, 243 pages. 2000.

Vol. 1760: J.-J. Ch. Meyer, P.-Y. Schobbens (Eds.), Formal Models of Agents. Poceedings. VIII, 253 pages. 1999. (Subseries LNAI).

Vol. 1761: R. Cafera, G. Salzer (Eds.), Automated Deduction in Classical and Non-Classical Logics. Proceedings. VIII, 299 pages. 2000. (Subseries LNAI).

Vol. 1762: K.-D. Schewe, B. Thalheim (Eds.), Foundations of Information and Knowledge Systems. Proceedings, 2000. X, 305 pages. 2000.

Vol. 1763: J. Akiyama, M. Kano, M. Urabe (Eds.), Discrete and Computational Geometry. Proceedings, 1998. VIII, 333 pages. 2000.

Vol. 1767: G. Bongiovanni, G. Gambosi, R. Petreschi (Eds.), Algorithms and Complexity. Proceedings, 2000. VIII, 317 pages. 2000.

Vol. 1768: A. Pfitzmann (Ed.), Information Hidig. Proceedings, 1999. IX, 492 pages. 2000.

Vol. 1769: G. Haring, C. Lindemann, M. Reiser (Eds.), Performance Evaluation: Origins and Directions. X, 529 pages. 2000.

Vol. 1770: H. Reichel, S. Tison (Eds.), STACS 2000. Proceedings, 2000. XIV, 662 pages. 2000.

Vol. 1774: J. Delgado, G.D. Stamoulis, A. Mullery, D. Prevedourou, K. Start (Eds.), Telecommunications and IT Convergence Towards Service E-volution. Proceedings, 2000. XIII, 350 pages. 2000.

Vol. 1780: R. Conradi (Ed.), Software Process Technology. Proceedings, 2000. IX, 249 pages. 2000.

Preface

SAC'99 was the sixth in a series of annual workshops on Selected Areas in Cryptography. Previous workshops were held at Carleton University in Ottawa (1995 and 1997) and at Queen's University in Kingston (1994, 1996, and 1998). The intent of the annual workshop is to provide a relaxed atmosphere in which researchers in cryptography can present and discuss new work on selected areas of current interest. The themes for the SAC'99 workshop were:

- Design and Analysis of Symmetric Key Cryptosystems
- Efficient Implementations of Cryptographic Systems
- Cryptographic Solutions for Web/Internet Security

The timing of the workshop was particularly fortuitous as the announcement by NIST of the five finalists for AES coincided with the first morning of the workshop, precipitating lively discussion on the merits of the selection!

A total of 29 papers were submitted to SAC'99 and, after a review process that had all papers reviewed by at least 3 referees, 17 were accepted and presented. As well, two invited presentations were given: one by Miles Smid from NIST entitled "From DES to AES: Twenty Years of Government Initiatives in Cryptography" and the other by Mike Reiter from Bell Labs entitled "Password Hardening with Applications to VPN Security".

The program committee for SAC'99 consisted of the following members: Carlisle Adams, Tom Cusick, Howard Heys, Lars Knudsen, Henk Meijer, Luke O'Connor, Doug Stinson, Stafford Tavares, and Serge Vaudenay. As well, additional reviewers were: Christian Cachin, Louis Granboulan, Helena Handschuh, Julio Lopez Hernandez, Mike Just, Alfred Menezes, Serge Mister, Guillaume Poupard, Victor Shoup, Michael Wiener, and Robert Zuccherato.

The organizers are very grateful for the financial support for the workshop received from Entrust Technologies, the Department of Electrical and Computer Engineering at Queen's University, and Communications and Information Technology Ontario (CITO). Special thanks to Stafford and Henk must be given for, once again, hosting SAC and being responsible for all the local arrangement details. The organizers would also like to thank Sheila Hutchison of the Department of Electrical and Computer Engineering at Queen's University for administrative and secretarial help and Yaser El-Sayed from the Faculty of Engineering at Memorial University of Newfoundland for help in preparing the workshop proceedings.

On behalf of the SAC'99 organizing committee, we thank all the workshop participants for making SAC'99 a success!

November 1999 Howard Heys and Carlisle Adams

Organization

Program Committee

Howard Heys (co-chair)	Memorial University of Newfoundland
Carlisle Adams (co-chair)	Entrust Technologies, Ottawa
Tom Cusick	SUNY, Buffalo
Lars Knudsen	University of Bergen
Henk Meijer	Queen's University at Kingston
Luke O'Connor	IBM, Zurich
Doug Stinson	University of Waterloo
Stafford Tavares	Queen's University at Kingston
Serge Vaudenay	Ecole Normale Supérieure, Paris

Local Organizing Committee

Stafford Tavares	Queen's University at Kingston
Henk Meijer	Queen's University at Kingston

Table of Contents

Cryptosystems and Pseudorandom Number Generators

Security Aspects of Block Ciphers

Cryptanalysis of Block Ciphers

Efficient Implementations of Cryptosystems

Cryptography for Network Applications

A Universal Encryption Standard

Helena Handschuh[1] and Serge Vaudenay[2]

[1] Gemplus - ENST
handschuh@gemplus.com
[2] Ecole Normale Supérieure - CNRS
Serge.Vaudenay@dmi.ens.fr

Abstract. DES and triple-DES are two well-known and popular encryption algorithms, but they both have the same drawback: their block size is limited to 64 bits. While the cryptographic community is working hard to select and evaluate candidates and finalists for the AES (Advanced Encryption Standard) contest launched by NIST in 1997, it might be of interest to propose a secure and simple double block-length encryption algorithm. More than in terms of key length and block size, our Universal Encryption Standard is a new construction that remains totally compliant with DES and triple-DES specifications as well as with AES requirements.

1 Introduction

For many years, DES [9] has been used as a worldwide encryption standard. But as technology improved for specialized key-search machines [26, 8], its 56-bit key size became too short, and a replacement was needed. 2-key triple-DES has since become the traditional block cipher used both by the cryptographic community as well as industry. However, there is a second drawback to DES which is also the case for triple-DES: its 64-bit block size. Therefore NIST launched a contest to select and evaluate candidates for a new encryption standard, the AES, in late 1997 [1]. The basic requirements for this new algorithm were that it be at least as secure and fast as triple-DES, but that its block size be of 128 bits instead of 64, and that its key size take possible values of 128, 192 and 256 bits.

Meanwhile, people are still using DES and triple-DES, and may want to start developping applications where these two as well as the new AES may independently be used as the encryption components. In order to be compliant with DES and triple-DES, we propose a new construction which is based on these building blocks, but which can take AES specifications as a requirement for its key and block sizes. Therefore, when AES is finally selected, it will come as a natural plug-in replacement of the actual structure whithout anybody being forced to change input and output interfaces.

We notice that double block-length encryption primitives based on DES already exist: as an example, take DEAL, which uses DES as the round function in a traditional 6-round Feistel scheme [16]. One can also think of multiple modes with

Howard Heys and Carlisle Adams (Eds.): SAC'99, LNCS 1758, pp. 1–12, 2000.

two blocks, where DES is the underlying cipher [10], but except for two-key triple DES in outer CBC mode which is vulnerable to dictionary and matching ciphertext attacks, none of these constructions are backward compliant with DES and triple-DES, nore do they make use of the full strength of a 128-bit block size (the second half of the plaintext never influences the first half of the ciphertext). Furthermore, multiple modes are either insecure [3–6] or require confidentiality or integrity protected initial values [25, 11]. We are also aware of the attacks by Lucks on 3-key triple DES [18] and DEAL [19].

The rest of the paper is organized as follows: section 2 presents our new encryption standard. Sections 3 and 4 provide details on collision attacks when some of the components of our UES are cut out. Section 5 provides additional security arguments on our construction and evaluates its strength based on the FX construction. Finally, we argue why we believe our construction is sound.

2 A Universal Encryption Standard

In this section we give the specifications of our new double block-length encryption algorithm. It basically runs two triple-DES encryptions in parallel and exchanges some of the bits of both halves inbetween each of the three encryption layers. Note that Outerbridge proposed a similar idea [21]. We investigated several related constructions and decided to add pre and post-whitening with extra keys, as well as an additional layer where bits of the left and the right half of the scheme are swapped under control of the extended secret key. Justification for these final choices will be given throughout this paper. The key schedule is considered to be the same as DEAL's.

2.1 Notations

We use the following notations for our scheme as well as for the attacks presented in the next sections (all operations are on bitstrings):

$a|b$: concatenation of a and b
$a \oplus b$: bitwise "exclusive or" of a and b
$a \wedge b$: bitwise "and" of a and b
$\overline{a}$: bitwise 1-complement of a
$\mathtt{001110100111}_\mathrm{b}$: bitstring in binary notation
$\mathtt{3a7}_\mathrm{x}$: bitstring in hexadecimal notation with implicit length (multiple of four)

In addition we let $\mathrm{DES}_k(x)$ denote the DES encryption of a 64-bit block x by using a 56-bit key k, and we let $3\mathrm{DES}_{k_1,k_2}(x)$ denote the 2-key triple-DES encryption of x in EDE mode (Encryption followed by Decryption followed by Encryption), *i.e.*

$$3\mathrm{DES}_{k_1,k_2}(x) = \mathrm{DES}_{k_1}\left(\mathrm{DES}_{k_2}^{-1}\left(\mathrm{DES}_{k_1}(x)\right)\right).$$

2.2 Basic Building Blocks

We already mentioned that we use parallel 3DES as well as a kind of keyed swap. In order to further formalize our proposal, let us define the following three basic building blocks which refer to operations on 128-bit strings. For convenience, we split a 128-bit string x into two 64-bit halves x_h and x_l.

1. **Keyed Translation.** Let $k = k_h|k_l$ be a 128-bit string. We define
$$T_k(x) = x \oplus k.$$
2. **Keyed Swap.** Let k be a 64-bit string. We define
$$S_k(x) = (x_h \oplus u)|(x_l \oplus u)$$
where $u = (x_h \oplus x_l) \wedge k$. This actually consists of exchanging the bits which are masked by k in the two halves.
3. **Parallel Encryption.** Let $k = k_h|k_l$ be two concatenated keys for two keyed algorithms C and C'. We define
$$P_{k,C,C'}(x) = C_{k_h}(x_h)|C'_{k_l}(x_l).$$

Our algorithm is a combination of three rounds of products of these transformations with additional operations before the first and after the last encryption layer.

2.3 Our New DES and 3DES-Compliant Construction

Having defined the above components, let $m = \texttt{00000000ffffffff}_\text{x}$, and let $k' = k_1|k_2|k_3|k_4$ and $m' = m_1|m_2|m_3|m_4$ be respectively two 256-bit extended keys derived from k by the key schedule.

Definition 1.

$$\text{UES}^*_k = P_{k_1|k_3,\text{DES},\text{DES}} \circ S_m \circ P_{k_2|k_4,\text{DES}^{-1},\text{DES}^{-1}} \circ S_m \circ P_{k_1|k_3,\text{DES},\text{DES}}$$

See figure 1. Then the precise formula to encrypt a plaintext under key k using UES reads as follows:

Definition 2.

$$\text{UES}_k = S_{m_4} \circ T_{m_3|m_3} \circ \text{UES}^*_k \circ T_{m_2|m_2} \circ S_{m_1}$$

See figure 2. This algorithm has two interesting properties. Namely if we set $m' = 0$ and $k' = k$, we have

Property 1.

$$\text{UES}_{k_1|k_2|k_1|k_2}(x_l|x_l) = \text{UES}^*_{k_1|k_2|k_1|k_2}(x_l|x_l) = 3\text{DES}_{k_1,k_2}(x_l)|3\text{DES}_{k_1,k_2}(x_l)$$

and

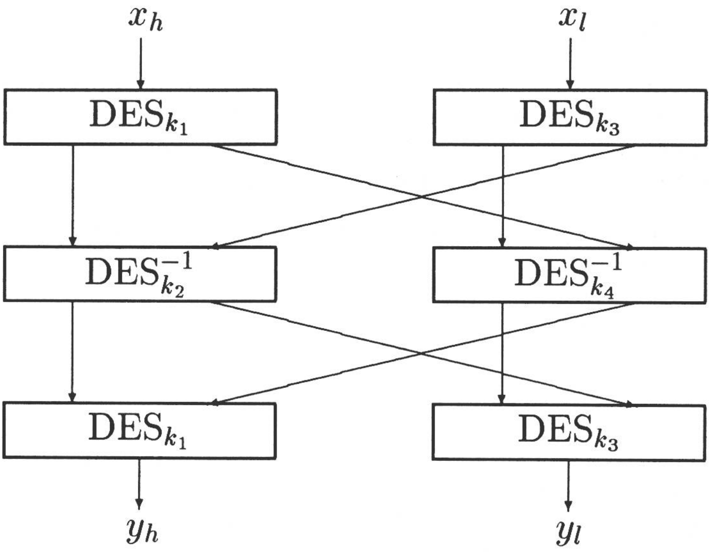

Fig. 1. UES*: Double-block length parallel triple DES

Property 2.

$$\mathrm{UES}_{k_1|k_1|k_1|k_1}(x_l|x_l) = \mathrm{UES}^*_{k_1|k_1|k_1|k_1}(x_l|x_l) = \mathrm{DES}_{k_1}(x_l)|\mathrm{DES}_{k_1}(x_l).$$

In addition it operates on 128-bit block messages. This makes the algorithm compatible with the forthcoming AES, and usable in DES or triple-DES mode. Finally, if we set $m = 0$, we can even run two full DES or 3DES encryptions in parallel, which doubles the encryption speed (two blocks are encrypted applying UES* only once).

Note that this scheme enables to construct double block-length encryption algorithms no matter what the underlying cipher is. For simplicity throughout this paper we will consider DES, but any other secure 64-bit block cipher could do the job. We will also focus on generic attacks that do not exploit the internal structure of the component encryption algorithm. Specific attacks such as differential [7] or linear cryptanalysis [20], truncated or higher order differentials [15] do not apply in this context as at least three layers of basic encryption are applied. We also believe that the best way to attack the scheme by a generic method is to try to create inner collisions.

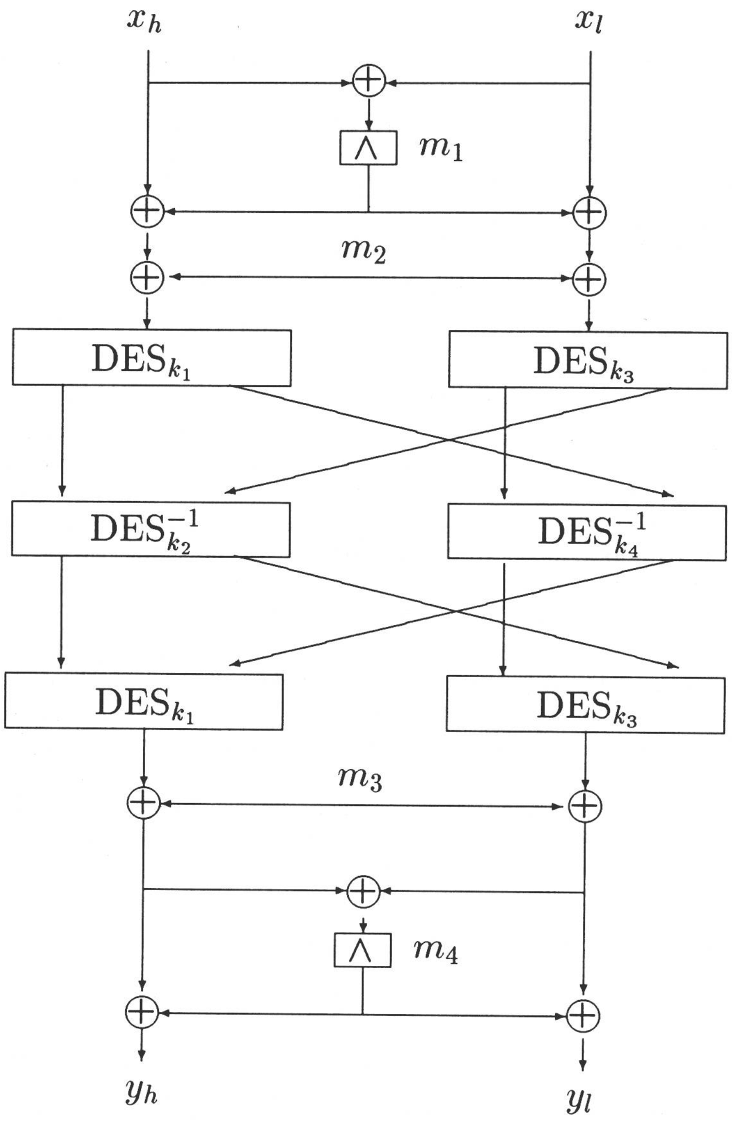

Fig. 2. Encryption with UES.

2.4 The Key-Schedule

In Table 1 below, we summarize in which different modes UES may be used.

Mode Key size	DES 56	3DES 112	AES 128/192/256
Block size 64 bits	$k' = k\vert k\vert k\vert k$ $m' = 0,\ m = 0$	$k' = k\vert k$ $m' = 0,\ m = 0$	-
Block size 128 bits	$k' = k\vert k\vert k\vert k$ $m' = 0, x_h = x_l$	$k' = k\vert k$ $m' = 0, x_h = x_l$	$k' = k_1\vert k_2\vert k_3\vert k_4$ $m' = m_1\vert m_2\vert m_3\vert m_4$

Table 1. Key-schedule for DES, 3DES and AES modes

The four subkeys and the four submasks used in AES-mode are derived from the user key using DEAL's key-schedule (for a 256-bit key). The user key is first divided into s subkeys of 64 bits each for $s = 2, 3, 4$.Then expand these s keys to 8 keys by repetition and exor the keys with a new constant for every repetition. Encrypt the expanded list of keys using DES in CBC mode with a fixed key $K = \texttt{0123456789abcdef}_{\mathrm{x}}$ and with the initial value set to zero. In order to partially allow on the fly key generation, start by deriving m_1 and m_2, next derive the four DES keys forming k', and finally derive m_3 and m_4.

We are aware of Kelsey and Schneier's [13] key-schedule cryptanalysis of DEAL. It turns out UES may have a very small class of equivalent keys in the 192-bit key case, because of the use of 56-bit keys for the inner DES blocks, whereas 64 bit subkeys are generated by the key-schedule. We also worked out a similar related-key attack with John Kelsey, which recovers the keys in complexity 2^{64} using 2^{33} related keys. However, these attacks apply in a very limited number of practical settings. Developpers should still make sure an attacker is not allowed to choose the keys in such a way.

3 Collision Attacks on Parallel DES

In this section, we consider the variant of UES previously defined as:

$$\mathrm{UES}^*_k = P_{k_1|k_3,\mathrm{DES},\mathrm{DES}} \circ S_m \circ P_{k_2|k_4,\mathrm{DES}^{-1},\mathrm{DES}^{-1}} \circ S_m \circ P_{k_1|k_3,\mathrm{DES},\mathrm{DES}}$$

We will show that this straightforward way of doubling the block size is not secure because a collision attack can be mounted against it (this phenomenon has been independently observed by Knudsen [17]). This is due to the fact that the construction is not a multipermutation. In other words, it may very well happen that if half of the input bits have a fixed value, half of the output bits also have, which would not be the case if the multipermutation property had been satisfied [22]. However, our intention is to prove that we can nevertheless use the structure if the input and output bits to this variant are unknown to the attacker. Therefore we begin by showing where the problem comes from, and justify our additional layers of swapping and masking in the final version of UES.