LNCS 4499 Journal Subline

Transactions on
Data Hiding and Multimedia Security II

Yun Q. Shi

Editor-in-Chief

Springer

Yun Q. Shi (Ed.)

Transactions on Data Hiding and Multimedia Security II

Springer

Volume Editor

Yun Q. Shi
New Jersey Institute of Technology
Department of Electrical and Computer Engineering
323, M.L. King Blvd., Newark, NJ 07102, USA
E-mail: shi@njit.edu

Library of Congress Control Number: 2007928444

CR Subject Classification (1998): K.4.1, K.6.5, H.5.1, D.4.6, E.3, E.4, F.2.2, H.3, I.4

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743 (Lecture Notes in Computer Science)
ISSN 1864-3043 (Transactions on Data Hiding and Multimedia Security)
ISBN-10 3-540-73091-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73091-0 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12077731 06/3180 5 4 3 2 1 0

Lecture Notes in Computer Science 4499

Lecture Notes in Computer Science

For information about Vols. 1–4447

please contact your bookseller or Springer

Vol. 4543: A.K. Bandara, M. Burgess (Eds.), Inter-Domain Management. XII, 237 pages. 2007.

Vol. 4542: P. Sawyer, B. Paech, P. Heymans (Eds.), Requirements Engineering: Foundation for Software Quality. IX, 384 pages. 2007.

Vol. 4541: T. Okadome, T. Yamazaki, M. Makhtari (Eds.), Pervasive Computing for Quality of Life Enhancemanet. IX, 248 pages. 2007.

Vol. 4539: N.H. Bshouty, C. Gentile (Eds.), Learning Theory. XII, 634 pages. 2007. (Sublibrary LNAI).

Vol. 4538: F. Escolano, M. Vento (Eds.), Graph-Based Representations in Pattern Recognition. XII, 416 pages. 2007.

Vol. 4537: K.C.-C. Chang, W. Wang, L. Chen, C.A. Ellis, C.-H. Hsu, A.C. Tsoi, H. Wang (Eds.), Advances in Web and Network Technologies, and Information Management. XXIII, 707 pages. 2007.

Vol. 4534: I. Tomkos, F. Neri, J. Solé Pareta, X. Masip Bruin, S. Sánchez Lopez (Eds.), Optical Network Design and Modeling. XI, 460 pages. 2007.

Vol. 4531: J. Indulska, K. Raymond (Eds.), Distributed Applications and Interoperable Systems. XI, 337 pages. 2007.

Vol. 4530: D.H. Akehurst, R. Vogel, R.F. Paige (Eds.), Model Driven Architecture- Foundations and Applications. X, 219 pages. 2007.

Vol. 4529: P. Melin, O. Castillo, L.T. Aguilar, J. Kacprzyk, W. Pedrycz (Eds.), Foundations of Fuzzy Logic and Soft Computing. XIX, 830 pages. 2007. (Sublibrary LNAI).

Vol. 4528: J. Mira, J.R. Álvarez (Eds.), Nature Inspired Problem-Solving Methods in Knowledge Engineering, Part II. XXII, 650 pages. 2007.

Vol. 4527: J. Mira, J.R. Álvarez (Eds.), Bio-inspired Modeling of Cognitive Tasks, Part I. XXII, 630 pages. 2007.

Vol. 4526: M. Malek, M. Reitenspieß, A. van Moorsel (Eds.), Service Availability. X, 155 pages. 2007.

Vol. 4525: C. Demetrescu (Ed.), Experimental Algorithms. XIII, 448 pages. 2007.

Vol. 4524: M. Marchiori, J.Z. Pan, C.d.S. Marie (Eds.), Web Reasoning and Rule Systems. XI, 382 pages. 2007.

Vol. 4523: Y.-H. Lee, H.-N. Kim, J. Kim, Y. Park, L.T. Yang, S.W. Kim (Eds.), Embedded Software and Systems. XIX, 829 pages. 2007.

Vol. 4522: B.K. Ersbøll, K.S. Pedersen (Eds.), Image Analysis. XVIII, 989 pages. 2007.

Vol. 4521: J. Katz, M. Yung (Eds.), Applied Cryptography and Network Security. XIII, 498 pages. 2007.

Vol. 4519: E. Franconi, M. Kifer, W. May (Eds.), The Semantic Web: Research and Applications. XVIII, 830 pages. 2007.

Vol. 4517: F. Boavida, E. Monteiro, S. Mascolo, Y. Koucheryavy (Eds.), Wired/Wireless Internet Communications. XIV, 382 pages. 2007.

Vol. 4516: L. Mason, T. Drwiega, J. Yan (Eds.), Managing Traffic Performance in Converged Networks. XXIII, 1191 pages. 2007.

Vol. 4515: M. Naor (Ed.), Advances in Cryptology - EUROCRYPT 2007. XIII, 591 pages. 2007.

Vol. 4514: S.N. Artemov, A. Nerode (Eds.), Logical Foundations of Computer Science. XI, 513 pages. 2007.

Vol. 4513: M. Fischetti, D.P. Williamson (Eds.), Integer Programming and Combinatorial Optimization. IX, 500 pages. 2007.

Vol. 4510: P. Van Hentenryck, L. Wolsey (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. X, 391 pages. 2007.

Vol. 4509: Z. Kobti, D. Wu (Eds.), Advances in Artificial Intelligence. XII, 552 pages. 2007. (Sublibrary LNAI).

Vol. 4508: M.-Y. Kao, X.-Y. Li (Eds.), Algorithmic Aspects in Information and Management. VIII, 428 pages. 2007.

Vol. 4507: F. Sandoval, A. Prieto, J. Cabestany, M. Graña (Eds.), Computational and Ambient Intelligence. XXVI, 1167 pages. 2007.

Vol. 4506: D. Zeng, I. Gotham, K. Komatsu, C. Lynch, M. Thurmond, D. Madigan, B. Lober, J. Kvach, H. Chen (Eds.), Intelligence and Security Informatics: Biosurveillance. XI, 234 pages. 2007.

Vol. 4505: G. Dong, X. Lin, W. Wang, Y. Yang, J.X. Yu (Eds.), Advances in Data and Web Management. XXII, 896 pages. 2007.

Vol. 4504: J. Huang, R. Kowalczyk, Z. Maamar, D. Martin, I. Müller, S. Stoutenburg, K.P. Sycara (Eds.), Service-Oriented Computing: Agents, Semantics, and Engineering. X, 175 pages. 2007.

Vol. 4501: J. Marques-Silva, K.A. Sakallah (Eds.), Theory and Applications of Satisfiability Testing – SAT 2007. XI, 384 pages. 2007.

Vol. 4500: N. Streitz, A. Kameas, I. Mavrommati (Eds.), The Disappearing Computer. XVIII, 304 pages. 2007.

Vol. 4499: Y.Q. Shi (Ed.), Transactions on Data Hiding and Multimedia Security II. IX, 117 pages. 2007.

Vol. 4497: S.B. Cooper, B. Löwe, A. Sorbi (Eds.), Computation and Logic in the Real World. XVIII, 826 pages. 2007.

Vol. 4496: N.T. Nguyen, A. Grzech, R.J. Howlett, L.C. Jain (Eds.), Agent and Multi-Agent Systems: Technologies and Applications. XXI, 1046 pages. 2007. (Sublibrary LNAI).

Vol. 4495: J. Krogstie, A. Opdahl, G. Sindre (Eds.), Advanced Information Systems Engineering. XVI, 606 pages. 2007.

Vol. 4494: H. Jin, O.F. Rana, Y. Pan, V.K. Prasanna (Eds.), Algorithms and Architectures for Parallel Processing. XIV, 508 pages. 2007.

Vol. 4493: D. Liu, S. Fei, Z. Hou, H. Zhang, C. Sun (Eds.), Advances in Neural Networks – ISNN 2007, Part III. XXVI, 1215 pages. 2007.

Vol. 4492: D. Liu, S. Fei, Z. Hou, H. Zhang, C. Sun (Eds.), Advances in Neural Networks – ISNN 2007, Part II. XXVII, 1321 pages. 2007.

Vol. 4491: D. Liu, S. Fei, Z.-G. Hou, H. Zhang, C. Sun (Eds.), Advances in Neural Networks – ISNN 2007, Part I. LIV, 1365 pages. 2007.

Vol. 4490: Y. Shi, G.D. van Albada, J. Dongarra, P.M.A. Sloot (Eds.), Computational Science – ICCS 2007, Part IV. XXXVII, 1211 pages. 2007.

Vol. 4489: Y. Shi, G.D. van Albada, J. Dongarra, P.M.A. Sloot (Eds.), Computational Science – ICCS 2007, Part III. XXXVII, 1257 pages. 2007.

Vol. 4488: Y. Shi, G.D. van Albada, J. Dongarra, P.M.A. Sloot (Eds.), Computational Science – ICCS 2007, Part II. XXXV, 1251 pages. 2007.

Vol. 4487: Y. Shi, G.D. van Albada, J. Dongarra, P.M.A. Sloot (Eds.), Computational Science – ICCS 2007, Part I. LXXXI, 1275 pages. 2007.

Vol. 4486: M. Bernardo, J. Hillston (Eds.), Formal Methods for Performance Evaluation. VII, 469 pages. 2007.

Vol. 4485: F. Sgallari, A. Murli, N. Paragios (Eds.), Scale Space and Variational Methods in Computer Vision. XV, 931 pages. 2007.

Vol. 4484: J.-Y. Cai, S.B. Cooper, H. Zhu (Eds.), Theory and Applications of Models of Computation. XIII, 772 pages. 2007.

Vol. 4483: C. Baral, G. Brewka, J. Schlipf (Eds.), Logic Programming and Nonmonotonic Reasoning. IX, 327 pages. 2007. (Sublibrary LNAI).

Vol. 4482: A. An, J. Stefanowski, S. Ramanna, C.J. Butz, W. Pedrycz, G. Wang (Eds.), Rough Sets, Fuzzy Sets, Data Mining and Granular Computing. XIV, 585 pages. 2007. (Sublibrary LNAI).

Vol. 4481: J. Yao, P. Lingras, W.-Z. Wu, M. Szczuka, N.J. Cercone, D. Ślęzak (Eds.), Rough Sets and Knowledge Technology. XIV, 576 pages. 2007. (Sublibrary LNAI).

Vol. 4480: A. LaMarca, M. Langheinrich, K.N. Truong (Eds.), Pervasive Computing. XIII, 369 pages. 2007.

Vol. 4479: I.F. Akyildiz, R. Sivakumar, E. Ekici, J.C.d. Oliveira, J. McNair (Eds.), NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet. XXVII, 1252 pages. 2007.

Vol. 4478: J. Martí, J.M. Benedí, A.M. Mendonça, J. Serrat (Eds.), Pattern Recognition and Image Analysis, Part II. XXVII, 657 pages. 2007.

Vol. 4477: J. Martí, J.M. Benedí, A.M. Mendonça, J. Serrat (Eds.), Pattern Recognition and Image Analysis, Part I. XXVII, 625 pages. 2007.

Vol. 4476: V. Gorodetsky, C. Zhang, V.A. Skormin, L. Cao (Eds.), Autonomous Intelligent Systems: Multi-Agents and Data Mining. XIII, 323 pages. 2007. (Sublibrary LNAI).

Vol. 4475: P. Crescenzi, G. Prencipe, G. Pucci (Eds.), Fun with Algorithms. X, 273 pages. 2007.

Vol. 4474: G. Prencipe, S. Zaks (Eds.), Structural Information and Communication Complexity. XI, 342 pages. 2007.

Vol. 4472: M. Haindl, J. Kittler, F. Roli (Eds.), Multiple Classifier Systems. XI, 524 pages. 2007.

Vol. 4471: P. Cesar, K. Chorianopoulos, J.F. Jensen (Eds.), Interactive TV: a Shared Experience. XIII, 236 pages. 2007.

Vol. 4470: Q. Wang, D. Pfahl, D.M. Raffo (Eds.), Software Process Dynamics and Agility. XI, 346 pages. 2007.

Vol. 4469: K.-C. Hui, Z. Pan, R.C.-k. Chung, C.C.L. Wang, X. Jin, S. Göbel, E.C.-L. Li (Eds.), Technologies for E-Learning and Digital Entertainment. XVIII, 974 pages. 2007.

Vol. 4468: M.M. Bonsangue, E.B. Johnsen (Eds.), Formal Methods for Open Object-Based Distributed Systems. X, 317 pages. 2007.

Vol. 4467: A.L. Murphy, J. Vitek (Eds.), Coordination Models and Languages. X, 325 pages. 2007.

Vol. 4466: F.B. Sachse, G. Seemann (Eds.), Functional Imaging and Modeling of the Heart. XV, 486 pages. 2007.

Vol. 4465: T. Chahed, B. Tuffin (Eds.), Network Control and Optimization. XIII, 305 pages. 2007.

Vol. 4464: E. Dawson, D.S. Wong (Eds.), Information Security Practice and Experience. XIII, 361 pages. 2007.

Vol. 4463: I. Măndoiu, A. Zelikovsky (Eds.), Bioinformatics Research and Applications. XV, 653 pages. 2007. (Sublibrary LNBI).

Vol. 4462: D. Sauveron, K. Markantonakis, A. Bilas, J.-J. Quisquater (Eds.), Information Security Theory and Practices. XII, 255 pages. 2007.

Vol. 4459: C. Cérin, K.-C. Li (Eds.), Advances in Grid and Pervasive Computing. XVI, 759 pages. 2007.

Vol. 4453: T. Speed, H. Huang (Eds.), Research in Computational Molecular Biology. XVI, 550 pages. 2007. (Sublibrary LNBI).

Vol. 4452: M. Fasli, O. Shehory (Eds.), Agent-Mediated Electronic Commerce. VIII, 249 pages. 2007. (Sublibrary LNAI).

Vol. 4451: T.S. Huang, A. Nijholt, M. Pantic, A. Pentland (Eds.), Artifical Intelligence for Human Computing. XVI, 359 pages. 2007. (Sublibrary LNAI).

Vol. 4450: T. Okamoto, X. Wang (Eds.), Public Key Cryptography – PKC 2007. XIII, 491 pages. 2007.

Vol. 4448: M. Giacobini et al. (Ed.), Applications of Evolutionary Computing. XXIII, 755 pages. 2007.

Preface

In this volume we present the second issue of the *LNCS Transactions on Data Hiding and Multimedia Security.*

In the first paper, Adelsbach et al. introduce fingercasting, a combination of broadcast encryption and fingerprinting for secure content distribution. They also provide for the first time a security proof for a lookup table-based encryption scheme. In the second paper, He and Kirovski propose an estimation attack on content-based video fingerprinting schemes. Although the authors tailor the attack towards a specific video fingerprint, the generic form of the attack is expected to be applicable to a wide range of video watermarking schemes. In the third paper, Ye et al. present a new feature distance measure for error-resilient image authentication, which allows one to differentiate maliciousimage manipulations from changes that do not interfere with the semantics of an image. In the fourth paper, Luo et al. present a steganalytic technique against steganographic embedding methods utilizing the two least significant bit planes. Experimental results demonstrate that this steganalysis method can reliably detect embedded messages and estimate their length with high precision. Finally, Alface and Macq present a comprehensive survey on blind and robust 3-D shape watermarking.

We hope that this issue is of great interest to the research community and will trigger new research in the field of data hiding and multimedia security.

Finally, we want to thank all the authors, reviewers and editors who devoted their valuable time to the success of this second issue. Special thanks go to Springer and Alfred Hofmann for their continuous support.

March 2007

Yun Q. Shi
(Editor-in-Chief)
Hyoung-Joong Kim
(Vice Editor-in-Chief)
Stefan Katzenbeisser
(Vice Editor-in-Chief)

LNCS Transactions on *Data Hiding and Multimedia Security*

Editorial Board

Advisory Board

Table of Contents

Fingercasting–Joint Fingerprinting and Decryption of Broadcast Messages 1
André Adelsbach, Ulrich Huber, and Ahmad-Reza Sadeghi

An Estimation Attack on Content-Based Video Fingerprinting 35
Shan He and Darko Kirovski

Statistics- and Spatiality-Based Feature Distance Measure for Error Resilient Image Authentication 48
Shuiming Ye, Qibin Sun, and Ee-Chien Chang

LTSB Steganalysis Based on Quartic Equation 68
Xiangyang Luo, Chunfang Yang, Daoshun Wang, and Fenlin Liu

From 3D Mesh Data Hiding to 3D Shape Blind and Robust Watermarking: A Survey 91
Patrice Rondao Alface and Benoit Macq

Author Index 117

Fingercasting–Joint Fingerprinting and Decryption of Broadcast Messages*

André Adelsbach, Ulrich Huber, and Ahmad-Reza Sadeghi

Horst Görtz Institute for IT Security
Ruhr-Universität Bochum
Universitätsstraße 150
D-44780 Bochum
Germany
andre.adelsbach@nds.rub.de, {huber,sadeghi}@crypto.rub.de

Abstract. We propose a stream cipher that provides confidentiality, traceability and renewability in the context of broadcast encryption assuming that collusion-resistant watermarks exist. We prove it to be as secure as the generic pseudo-random sequence on which it operates. This encryption approach, termed fingercasting, achieves joint decryption and fingerprinting of broadcast messages in such a way that an adversary cannot separate both operations or prevent them from happening simultaneously. The scheme is a combination of a known broadcast encryption scheme, a well-known class of fingerprinting schemes and an encryption scheme inspired by the Chameleon cipher. It is the first to provide a formal security proof and a non-constant lower bound for resistance against collusion of malicious users, i.e., a minimum number of content copies needed to remove all fingerprints. To achieve traceability, the scheme fingerprints the receivers' key tables such that they embed a fingerprint into the content during decryption. The scheme is efficient and includes parameters that allow, for example, to trade-off storage size for computation cost at the receiving end.

Keywords: Chameleon encryption, stream cipher, spread-spectrum watermarking, fingerprinting, collusion resistance, frame-proofness, broadcast encryption.

1 Introduction

Experience shows that adversaries attack Broadcast Encryption (BE) systems in a variety of different ways. Their attacks may be on the hardware that stores cryptographic keys, e.g., when they extract keys from a compliant device to develop a pirate device such as the DeCSS software that circumvents the Content Scrambling System [2]. Alternatively, their attacks may be on the decrypted content, e.g., when a legitimate user shares decrypted content with illegitimate users on a file sharing system such as Napster, Kazaa, and BitTorrent.

* An extended abstract of this paper appeared in the Proceedings of the Tenth Australasian Conference on Information Security and Privacy (ACISP 2006) [1].

Y.Q. Shi (Eds.): Transactions on DHMS II, LNCS 4499, pp. 1–34, 2007.

The broadcasting sender thus has three security requirements: *confidentiality*, *traceability* of content and keys, and *renewability* of the encryption scheme. The requirements cover two aspects. Confidentiality tries to prevent illegal copies in the first place, whereas traceability is a second line of defense aimed at finding the origin of an illegal copy (content or key). The need for traceability originates from the fact that confidentiality may be compromised in rare cases, e.g., when a few users illegally distribute their secret keys. Renewability ensures that after such rare events, the encryption system can recover from the security breach.

In broadcasting systems deployed today, e.g., Content Protection for Pre-Recorded Media [3] or the Advanced Access Content System [4], confidentiality and renewability often rely on BE because it provides short ciphertexts while at the same time having realistic storage requirements in devices and acceptable computational overhead. Traitor tracing enables traceability of keys, whereas fingerprinting provides traceability of content. Finally, renewability may be achieved using revocation of the leaked keys.

However, none of the mentioned cryptographic schemes covers all three security requirements. Some existing BE schemes lack traceability of keys, whereas no practically relevant scheme provides traceability of content [5,6,7,8]. Traitor tracing only provides traceability of keys, but not of content [9,10]. Fingerprinting schemes alone do not provide confidentiality [11]. The original Chameleon cipher provides confidentiality, traceability and a hint on renewability, but with a small constant bound for collusion resistance and, most importantly, without formal proof of security [12]. Asymmetric schemes, which provide each compliant device with a certificate and accompany content with Certificate Revocation Lists (CRLs), lack traceability of content and may reach the limits of renewability when CRLs become too large to be processed by real-world devices. Finally, a trivial combination of fingerprinting and encryption leads to an unacceptable transmission overhead because the broadcasting sender needs to sequentially transmit each fingerprinted copy.

Our Contribution. We present, to the best of our knowledge, the first rigorous security proof of Chameleon ciphers, thus providing a sound foundation for the recent applications of these ciphers, e.g., [13]. Furthermore, we give an explicit criterion to judge the security of the Chameleon cipher's key table. Our *fingercasting* approach fulfills all three security requirements at the same time. It is a combination of (i) a new Chameleon cipher based on the *finger*printing capabilities of a well-known class of watermarking schemes and (ii) an arbitrary broad*cast* encryption scheme, which explains the name of the approach. The basic idea is to use the Chameleon cipher for combining decryption and fingerprinting. To achieve renewability, we use a BE scheme to provide fresh session keys as input to the Chameleon scheme. To achieve traceability, we fingerprint the receivers' key tables such that they embed a fingerprint into the content during decryption. To enable higher collusion resistance than the original Chameleon scheme, we tailor our scheme to emulate any watermarking scheme whose coefficients follow a probability distribution that can be disaggregated into additive components.[1] As proof of concept, we instantiate the watermarking scheme with Spread Spectrum Watermarking (SSW), which has proven collusion resistance [14,15]. However, we might as well instantiate it with any other such scheme.

Joint decryption and fingerprinting has significant advantages compared to existing methods such as transmitter-side or receiver-side Fingerprint Embedding (FE) [11]. Transmitter-side FE is the trivial combination of fingerprinting and encryption by the sender. As discussed above, the transmission overhead is in the order of the number of copies to be distributed, which is prohibitive in practical applications. Receiver-side FE happens in the user's receiver; after distribution of a single encrypted copy of the content, a secure receiver based on tamper-resistant hardware is trusted to embed the fingerprint *after* decryption. This saves bandwidth on the broadcast channel. However, perfect tamper-resistance cannot be achieved under realistic assumptions [16]. An adversary may succeed in extracting the keys of a receiver and subsequently decrypt without embedding a fingerprint.

Our fingercasting approach combines the advantages of both methods. It saves bandwidth by broadcasting a single encrypted copy of the content. In addition, it ensures embedding of a fingerprint even if a malicious user succeeds in extracting the decryption keys of a receiver. Furthermore, as long as the number of colluding users remains below a threshold, the colluders can only create decryption keys and content copies that incriminate at least one of them.

This paper enhances our extended abstract [1] in the following aspects. First, the extended abstract does not contain the security proof, which is the major contribution. Second, we show here that our instantiation of SSW is exact, whereas the extended abstract only claims this result. Last, we discuss here the trade-off between storage size and computation cost at the receiving end.

2 Related Work

The original Chameleon cipher of Anderson and Manifavas is 3-collusion-resistant [12]: A collusion of up to 3 malicious users has a negligible chance of creating a good copy that does not incriminate them. Each legitimate user knows the seed of a Pseudo-Random Sequence (PRS) and a long table filled with random keywords. Based on the sender's master table, each receiver obtains a slightly different table copy, where individual bits in the keywords are modified in a characteristic way. Interpreting the PRS as a sequence of addresses in the table, the sender adds the corresponding keywords in the master table bitwise modulo 2 in order to mask the plaintext word. The receiver applies the same operation to the ciphertext using its table copy, thus embedding the fingerprint.

The original cipher, however, has some inconveniences. Most importantly, it has no formal security analysis and bounds the collusion resistance by the constant number 3, whereas our scheme allows to choose this bound depending on the number of available watermark coefficients. In addition, the original scheme

[1] Our scheme does not yet support fingerprints based on coding theory.

limits the content space (and keywords) to strings with characteristic bit positions that may be modified without visibly altering the content. In contrast, our scheme uses algebraic operations in a group of large order, which enables modification of any bit in the keyword and processing of arbitrary documents.

Chameleon was inspired by work from Maurer [17,18]. His cipher achieves information-theoretical security in the bounded storage model with high probability. In contrast, Chameleon and our proposed scheme only achieve computational security. The reason is that the master table length in Maurer's cipher is super-polynomial. As any adversary would need to store most of the table to validate guesses, the bounded storage capacity defeats all attacks with high probability. However, Maurer's cipher was never intended to provide traceability of content or renewability, but only confidentiality.

Ferguson et al. discovered security weaknesses in a randomized stream cipher similar to Chameleon [19]. However, their attack only works for linear sequences of keywords in the master table, not for the PRSs of our proposed solution.

Ergun, Kilian, and Kumar prove that an averaging attack with additional Gaussian noise defeats any watermarking scheme [20]. Their bound on the minimum number of different content copies needed for the attack asymptotically coincides with the bound on the maximum number of different content copies to which the watermarking scheme of Kilian et al. is collusion-resistant [15]. As we can emulate [15] with our fingercasting approach, its collusion resistance is—at least asymptotically—the best we can hope for.

Recently there was a great deal of interest in joint fingerprinting and decryption [13,21,22,11,23]. Basically, we can distinguish three strands of work. The first strand of work applies Chameleon in different application settings. Briscoe et al. introduce Nark, which is an application of the original Chameleon scheme in the context of Internet multicast [13]. However, in contrast to our new Chameleon cipher they neither enhance Chameleon nor analyze its security. The second strand of work tries to achieve joint fingerprinting and decryption by either trusting network nodes to embed fingerprints (Watercasting in [21]) or doubling the size of the ciphertext by sending differently fingerprinted packets of content [22]. Our proposed solution neither relies on trusted network nodes nor increases the ciphertext size. The third strand of work proposes new joint fingerprinting and decryption processes, but at the price of replacing encryption with scrambling, which does not achieve indistinguishability of ciphertext and has security concerns [11,23]. In contrast, our new Chameleon cipher achieves indistinguishability of ciphertext.

3 Preliminaries

3.1 Notation

We recall some standard notations that will be used throughout the paper. First, we denote scalar objects with lower-case variables, e.g., o_1, and object tuples as well as roles with upper-case variables, e.g., X_1. When we summarize objects or roles in set notation, we use an upper-case calligraphic variable, e.g., $\mathcal{O} := \{o_1, o_2, \ldots\}$ or $\mathcal{X} := \{X_1, X_2, \ldots\}$. Second, let A be an algorithm. By $y \leftarrow \mathsf{A}(x)$ we denote that y was obtained by running A on input x. If A is deterministic, then y is a variable with a unique value. Conversely, if A is probabilistic, then y is a random variable. For example, by $y \leftarrow \mathsf{N}(\mu, \sigma)$ we denote that y was obtained by selecting it at random with normal distribution, where μ is the mean and σ the standard deviation. Third, $o_1 \overset{R}{\leftarrow} \mathcal{O}$ and $o_2 \overset{R}{\leftarrow} [0, z]$ denote the selection of a random element of the set $\mathcal{O}$ and the interval $[0, z]$ with uniform distribution. Finally, $V \cdot W$ denotes the dot product of two vectors $V := (v_1, \ldots, v_n)$ and $W := (w_1, \ldots, w_n)$, which is defined as $V \cdot W := \sum_{j=1}^{n} v_j w_j$, while $||V||$ denotes the Euclidean norm $||V|| := \sqrt{V \cdot V}$.

3.2 Roles and Objects in Our System Model

The *(broadcast) center* manages the broadcast channel, distributes decryption keys and is fully trusted. The *users* obtain the content via devices that we refer to as *receivers*. For example, a receiver may be a set-top box in the context of pay-TV or a DVD player in movie distribution. We denote the number of receivers with N; the set of receivers is $\mathcal{U} := \{u_i \mid 1 \leq i \leq N\}$. When a receiver violates the terms and conditions of the application, e.g., leaks its keys or shares content, the center revokes the receiver's keys and thus makes them useless for decryption purposes. We denote the set of revoked receivers with $\mathcal{R} := \{r_1, r_2, \ldots\} \subset \mathcal{U}$.

We represent broadcast content as a sequence $M := (m_1, \ldots, m_n)$ of real numbers in $[0, z]$, where M is an element of the content space $\mathcal{M}$.[2] For example, these numbers may be the n most significant coefficients of the Discrete Cosine Transform (DCT) as described in [14]. However, they should not be thought of as a literal description of the underlying content, but as a representation of the values that are to be changed by the watermarking process [20]. We refer to these values as *significant* and to the remainder as *insignificant*. In the remainder of this paper, we only refer to the significant part of the content, but briefly comment on the insignificant part in Section 5.

3.3 Cryptographic Building Blocks

Negligible Function. A negligible function $\mathsf{f} : \mathbb{N} \rightarrow \mathbb{R}$ is a function where the inverse of any polynomial is asymptotically an upper bound:

$$\forall k > 0 \, \exists \lambda_0 \, \forall \lambda > \lambda_0 : \quad \mathsf{f}(\lambda) < 1/\lambda^k$$

Probabilistic Polynomial Time. A probabilistic polynomial-time algorithm is an algorithm for which there exists a polynomial poly such that for every input $x \in \{0, 1\}^*$ the algorithm always halts after $\mathsf{poly}(|x|)$ steps, independently of the outcome of its internal coin tosses.

[2] Although this representation mainly applies to images, we discuss an extension to movies and songs in Section 5.

Pseudo-Random Sequence (PRS). We first define the notion of pseudo-randomness and then proceed to define a Pseudo-Random Sequence Generator (PRSG). For further details we refer to [24, Section 3.3.1]:

Definition 1 (Pseudo-randomness). *Let* $\mathsf{len} : \mathbb{N} \rightarrow \mathbb{N}$ *be a polynomial such that* $\mathsf{len}(\lambda) > \lambda$ *for all* $\lambda \in \mathbb{N}$ *and let* $U_{\mathsf{len}(\lambda)}$ *be a random variable uniformly distributed over the strings* $\{0,1\}^{\mathsf{len}(\lambda)}$ *of length* $\mathsf{len}(\lambda)$. *Then the random variable* X *with* $|X| = \mathsf{len}(\lambda)$ *is called* pseudo-random *if for every probabilistic polynomial-time distinguisher* $\mathcal{D}$, *the advantage* $\mathsf{Adv}(\lambda)$ *is a negligible function:*

$$\mathsf{Adv}(\lambda) := \left|\mathsf{Pr}\left[\mathcal{D}(X) = 1\right] - \mathsf{Pr}\left[\mathcal{D}(U_{\mathsf{len}(\lambda)}) = 1\right]\right|$$

Definition 2 (Pseudo-Random Sequence Generator). *A* PRSG *is a deterministic polynomial-time algorithm* G *that satisfies two requirements:*

1. *Expansion: There exists a polynomial* $\mathsf{len} : \mathbb{N} \rightarrow \mathbb{N}$ *such that* $\mathsf{len}(\lambda) > \lambda$ *for all* $\lambda \in \mathbb{N}$ *and* $|\mathsf{G}(str)| = \mathsf{len}(|str|)$ *for all* $str \in \{0,1\}^*$.
2. *Pseudo-randomness: The random variable* $\mathsf{G}(U_\lambda)$ *is pseudo-random.*

A PRS *is a sequence* $\mathsf{G}(str)$ *derived from a uniformly distributed random seed* str *using a PRSG.*

Chameleon Encryption. To set up a Chameleon scheme $\mathcal{CE} := (\mathsf{KeyGenCE}, \mathsf{KeyExtrCE}, \mathsf{EncCE}, \mathsf{DecCE}, \mathsf{DetectCE})$, the center generates the secret master table MT, the secret table fingerprints $TF := (TF^{(1)}, \ldots, TF^{(N)})$, and selects a threshold t using the key generation algorithm $(MT, TF, t) \leftarrow \mathsf{KeyGenCE}(N, 1^{\lambda'}, par_{\mathsf{CE}})$, where N is the number of receivers, λ' a security parameter, and par_{CE} a set of performance parameters. To add receiver u_i to the system, the center uses the key extraction algorithm $RT^{(i)} \leftarrow \mathsf{KeyExtrCE}(MT, TF, i)$ to deliver the secret receiver table $RT^{(i)}$ to u_i. To encrypt content M exclusively for the receivers in possession of a receiver table $RT^{(i)}$ and a fresh session key k^{sess}, the center uses the encryption algorithm $C \leftarrow \mathsf{EncCE}(MT, k^{\mathsf{sess}}, M)$, where the output is the ciphertext C. Only a receiver u_i in possession of $RT^{(i)}$ and k^{sess} is capable of decrypting C and obtaining a fingerprinted copy $M^{(i)}$ of content M using the decryption algorithm $M^{(i)} \leftarrow \mathsf{DecCE}(RT^{(i)}, k^{\mathsf{sess}}, C)$.

When the center discovers an illegal copy M^* of content M, it executes $\mathsf{DetectCE}$, which uses the fingerprint detection algorithm $\mathsf{DetectFP}$ of the underlying fingerprinting scheme to detect whether $RT^{(i)}$ left traces in M^*. For further details on our notation of a Chameleon scheme, we refer to Appendix C.

Fingerprinting. To set up a fingerprinting scheme, the center generates the secret content fingerprints $CF := (CF^{(1)}, \ldots, CF^{(N)})$ and the secret similarity threshold t using the setup algorithm $(CF, t) \leftarrow \mathsf{SetupFP}(N, n', par_{\mathsf{FP}})$, where N is the number of receivers, n' the number of content coefficients, and par_{FP} a set of performance parameters. To embed the content fingerprint $CF^{(i)} := (cf_1^{(i)}, \ldots, cf_{n'}^{(i)})$ of receiver u_i into the original content M, the center uses the embedding algorithm $M^{(i)} \leftarrow \mathsf{EmbedFP}(M, CF^{(i)})$. To verify whether an illegal copy M^* of content M contains traces of the content fingerprint $CF^{(i)}$ of receiver

u_i, the center uses the detection algorithm $dec \leftarrow \mathsf{DetectFP}(M, M^*, CF^{(i)}, t)$. It calculates the similarity between the detected fingerprint $CF^* := M^* - M$ and $CF^{(i)}$ using a similarity measure. If the similarity is above the threshold t, then the center declares u_i guilty ($dec = \mathtt{true}$), otherwise innocent ($dec = \mathtt{false}$). This type of detection algorithm is called non-blind because it needs the original content M as input; the opposite is a blind detection algorithm.

We call a fingerprinting scheme *additive* if the probability distribution ProDis of its coefficients has the following property: Adding two independent random variables that follow ProDis results in a random variable that also follows ProDis. For example, the normal distribution has this property, where the means and variances add up during addition.

Spread Spectrum Watermarking (SSW) is an instance of an additive fingerprinting scheme. We describe the SSW scheme of [15], which we later use to achieve collusion resistance. The content fingerprint $CF^{(i)}$ consists of independent random variables $cf_j^{(i)}$ with normal distribution $\mathsf{ProDis} = \mathsf{N}(0, \sigma')$, where σ' is a function $\mathsf{f}_{\sigma'}(N, n', par_{\mathsf{FP}})$. The similarity threshold t is a function $\mathsf{f}_t(\sigma', N, par_{\mathsf{FP}})$. Both functions $\mathsf{f}_{\sigma'}$ and f_t are specified in [15]. During $\mathsf{EmbedFP}$, the center adds the fingerprint coefficients to the content coefficients: $m_j^{(i)} \leftarrow m_j + cf_j^{(i)}$. The similarity test is $\mathsf{Sim}(CF^*, CF^{(i)}) \geq t$ with $\mathsf{Sim}(CF^*, CF^{(i)}) := (CF^* \cdot CF^{(i)})/||CF^*||$. Finally, the scheme's security is given by:

Theorem 1. [15, Section 3.4] *In the SSW scheme with the above parameters, an adversarial coalition needs $\Omega(\sqrt{n'/\ln N})$ differently fingerprinted copies of content M to have a non-negligible chance of creating a good copy M^* without any coalition member's fingerprint.*

For further details on our notation of a fingerprinting scheme and the SSW scheme of [15], we refer to Appendix D.

Broadcast Encryption. To set up the scheme, the center generates the secret master key MK using the key generation algorithm $MK \leftarrow \mathsf{KeyGenBE}(N, 1^{\lambda''})$, where N is the number of receivers and λ'' the security parameter. To add receiver u_i to the system, the center uses the key extraction algorithm $SK^{(i)} \leftarrow \mathsf{KeyExtrBE}(MK, i)$ to extract the secret key $SK^{(i)}$ of u_i. To encrypt session key k^{sess} exclusively for the non-revoked receivers $\mathcal{U} \setminus \mathcal{R}$, the center uses the encryption algorithm $C \leftarrow \mathsf{EncBE}(MK, \mathcal{R}, k^{\mathsf{sess}})$, where the output is the ciphertext C. Only a non-revoked receiver u_i has a matching private key $SK^{(i)}$ that allows to decrypt C and obtain k^{sess} using the decryption algorithm $k^{\mathsf{sess}} \leftarrow \mathsf{DecBE}(i, SK^{(i)}, C)$. For further details on our notation of a BE scheme, we refer to Appendix E.

3.4 Requirements of a Fingercasting Scheme

Before we enter into the details of our fingercasting approach, we summarize its requirements: correctness, security, collusion resistance, and frame-proofness. To put it simply, the aim of our fingercasting approach is to generically combine an instance of a BE scheme, a Chameleon scheme, and a fingerprinting scheme