# 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, 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. We prove it to be as secure as the generic pseudo-random sequence on which it operates. This encryption scheme, termed fingercasting scheme, 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 broadcast encryption scheme, a fingerprinting scheme 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. The scheme is efficient and includes parameters that allow, for example, to trade-off storage size for computation cost at the receiving end.

## 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 [1]. 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.

The broadcasting sender thus has three security requirements: *confidentiality*, *traceability* of content and keys, and *renewability* of the encryption scheme. Confidentiality tries to prevent illegal copies, whereas traceability is a second line of defense aimed at finding the origin of an illegal copy. The need for traceability implies 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., CPPM [2] or AACS [3], 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,

L. Batten and R. Safavi-Naini (Eds.): ACISP 2006, LNCS 4058, pp. 136–147, 2006.

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 BE schemes lack traceability of keys, whereas no practically relevant scheme provides traceability of content [4, 5, 6, 7]. Traitor tracing does not provide traceability of of content [8, 9]. Fingerprinting schemes do not provide confidentiality [10]. The original Chameleon cipher provides confidentiality, traceability and a hint on renewability, but with a small constant bound for collusion resistance and without formal proof of security [11]. Asymmetric schemes, which provide each compliant device with a certificate and accompany content with Certificate Revocation Lists (CRLs), lack traceability of content and reach the limits of renewability when CRLs become too large. A trivial combination of fingerprinting and encryption leads to an unacceptable transmission overhead because the sender needs to sequentially transmit each fingerprinted copy. Finally, receiver-side fingerprint embedding relies on the tamper resistance of the receivers' hardware [10], which is doubtful in practice. Each receiver is trusted to embed the fingerprint *after* decryption. However, perfect tamper-resistance cannot be achieved under realistic assumptions [15].

We present, to the best of our knowledge, the first rigorous security proof of Chameleon ciphers, thus providing a sound foundation for their recent applications, e.g., [12]. 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 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 cipher. 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 cipher to emulate any watermarking scheme whose coefficients can be disaggregated into additive components. As proof of concept, we instantiate the watermarking scheme with Spread Spectrum Watermarking (SSW), which has proven collusion resistance [13, 14]. However, we might as well use any other such scheme.

We note that our fingercasting approach distributes 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 his receiver. As long as the number of colluding users remains below a threshold, they can only create new decryption keys and content copies that incriminate at least one of them.

## 2 Related Work

The original Chameleon cipher of Anderson and Manifavas is a 3-collusion-resistant fingercasting scheme [11]: A collusion of up to 3 malicious users has a negligible chance of creating of 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 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 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 [16]. 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. 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 [17]. 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 [18]. 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 [14]. As we emulate [14], its collusion resistance is asymptotically the best we can hope for.

Recently there was a great deal of interest in joint fingerprinting and decryption [12, 19, 20, 10, 21]. 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 [12]. However, in contrast to our new Chameleon scheme they neither enhance the original scheme 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 [19]) or doubling the size of the ciphertext by sending differently fingerprinted packets of content [20]. 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 [10, 21]. In contrast, our new Chameleon scheme 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 $\mathsf{A}$ be an algorithm. By $y \leftarrow \mathsf{A}(x)$ we denote that $y$ was obtained by running $\mathsf{A}$ on input $x$. For example, by $y \leftarrow \mathsf{N}(\mu, \sigma)$ we denote that $y$ was obtained by selecting it at random with normal distribution, where $\mu$ is the mean and $\sigma$ the standard deviation. Third, $o_1 \xleftarrow{R} \mathcal{O}$ and $o_2 \xleftarrow{R} [0, z]$ 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 $\mathcal{M}$. For example, these numbers may be the $n$ most significant coefficients of the Discrete Cosine Transform (DCT) of an image as described in [13]. 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 [18].

## 3.3 Cryptographic Building Blocks

**Chameleon Encryption.** To set up the scheme $\mathcal{CE} := (\mathsf{KeyGenCE}, \mathsf{KeyExtrCE}, \mathsf{EncCE}, \mathsf{DecCE})$, the center generates the secret master table $MT$ and the secret table fingerprints $TF := (TF^{(1)}, \ldots, TF^{(N)})$ using the key generation algorithm $(MT, TF) \leftarrow \mathsf{KeyGenCE}(N, 1^{\lambda'}, par_{\mathsf{CE}})$, where $N$ is the number of receivers, $\lambda'$ a security parameter, and $par_{\mathsf{CE}}$ a set of 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 session key $k^{\mathsf{sess}}$, 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 $k^{\mathsf{sess}}$ is capable of decrypting $C$ and obtaining a fingerprinted copy $M^{(i)}$ of $M$ 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 uses the fingerprint detection algorithm of the underlying fingerprinting scheme.

**Fingerprinting.** To set up the 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 $par_{\mathsf{FP}}$ 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 ($dec = \mathtt{false}$). The detection algorithm is called non-blind because it needs the original content $M$ as input.

In the sequel, we call a fingerprinting scheme *additive* if the probability distribution $\mathsf{Prob}$ of its coefficients has the following property: Adding two independent random variables that follow $\mathsf{Prob}$ results in a random variable that also follows $\mathsf{Prob}$. Spread Spectrum Watermarking (SSW) is an instance of an additive fingerprinting scheme [14]. The content fingerprint $CF^{(i)}$ consists of independent random variables $cf_j^{(i)}$ with distribution $\mathsf{Prob} = \mathsf{N}(0, \sigma')$, where $\sigma'$ 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 $\mathsf{f}_t$ are specified in [14]. During $\mathsf{EmbedFP}$, the center adds the fingerprint coefficients to the content coefficients: $m_j^{(i)} \leftarrow m_j + cf_j^{(i)}$. The similarity measure is $\mathsf{Sim}(CF^*, CF^{(i)}) := (CF^* \cdot CF^{(i)})/||CF^*||$.

**Theorem 1.** [14, 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.*

**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 $1^{\lambda''}$ 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 content $M$ exclusively for the non-revoked receivers $\mathcal{U} \setminus \mathcal{R}$, the center uses the encryption algorithm $C \leftarrow \mathsf{EncBE}(MK, \mathcal{R}, M)$, 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 $M$ using the decryption algorithm $M \leftarrow \mathsf{DecBE}(i, SK^{(i)}, C)$.

**Pseudo-random Sequence (PRS).** We formally define the term PRS in the technical report [22]. Informally, a PRS is a long bit string that no efficient algorithm can distinguish from a truly random bit string of identical length. In

order to create a PRS, a Pseudo-Random Sequence Generator (PRSG), which is a deterministic polynomial-time algorithm $\mathsf{G}$, derives the long bit string from a short random seed. If $|\sigma| \in \mathbb{N}$ is the length of the seed $\sigma$, then the expansion factor $\mathsf{len} : \mathbb{N} \to \mathbb{N}$ of the PRSG determines the length of the PRS: $|\mathsf{G}(\sigma)| = \mathsf{len}(|\sigma|)$.

### 3.4 Requirements of a Fingercasting Scheme

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

We informally explain the requirements one by one; a formal definition is available in [22]. Correctness requires that the receiver obtains with high probability a fingerprinted copy that is perceptually indistinguishable from the original; in other words, the fingerprint may not deteriorate the content and lead to a bad copy. We denote the residual probability of a bad copy with $p^{\mathsf{bad}}$, which obviously should be close to 0 for practical purposes. The SSW scheme of [14] uses the measure $||M^{(i)} - M|| \leq \sqrt{n'}\delta$ to decide whether a copy is good, where $n'$ is the number of content coefficients and $\delta$ a goodness criterion.

All relevant BE schemes provide even stronger security notions than Chameleon [5, 6, 7]. The remaining requirements therefore only relate to the Chameleon scheme $\mathcal{CE}$. We informally define security of $\mathcal{CE}$ as follows: No efficient algorithm may be capable of distinguishing the ciphertexts of two messages, even if this algorithm selects the two messages to be encrypted and obtains an encryption of one of the two messages selected at random.

Resistance against a collusion of up to $q$ colluders (or $q$-collusion resistance) means that no efficient algorithm with access to the secret information of at most $q$ colluders may be capable of creating a good content copy such that the fingerprint detection algorithm incriminates none of the colluders. We denote the residual probability of a successful collusion with $p^{\mathsf{neg}}$, which is usually called the false negative probability. 1-collusion resistance is known as robustness.

Frame-proofness is similar to collusion resistance, but the adversarial coalition wins if the detection algorithm accuses an innocent user. We denote the residual probability of a successful framing attack with $p^{\mathsf{pos}}$, which is usually called the false positive probability.

## 4 Proposed Solution

### 4.1 High-Level Overview of the Proposed Fingercasting Scheme

To fingercast content, the center uses the BE scheme to send a fresh session key to each non-revoked receiver. This session key initializes a pseudo-random sequence generator. The resulting pseudo-random sequence represents a sequence

of addresses in the master table of our new Chameleon scheme. The center encrypts the content with the master table entries to which the addresses refer. Each receiver has a unique receiver table that differs only slightly from the master table. During decryption, these slight differences in the receiver table lead to slight, but characteristic differences in the content copy.

We divide this approach into the same five steps that we have seen for Chameleon schemes in Section 3.3. First, the *key generation* algorithm of the fingercasting scheme consists of the key generations algorithms of the two underlying schemes $\mathsf{KeyGenBE}$ and $\mathsf{KeyGenCE}$. The center's master key thus consists of $MK$, $MT$ and $TF$. Second, the same observation holds for the *key extraction* algorithm of the fingercasting scheme. It consists of the respective algorithms in the two underlying schemes $\mathsf{KeyExtrBE}$ and $\mathsf{KeyExtrCE}$. The secret key of receiver $u_i$ therefore has two elements: $SK^{(i)}$ and $RT^{(i)}$.

Third, the *encryption* algorithm defines how we interlock the two underlying schemes. To encrypt, the center generates a fresh and random session key $k^{\mathsf{sess}} \xleftarrow{R} \{0,1\}^{\lambda}$. This session key is broadcasted to the non-revoked receivers using the BE scheme: $C_{\mathsf{BE}} \leftarrow \mathsf{EncBE}(MK, \mathcal{R}, k^{\mathsf{sess}})$. Subsequently, the center uses $k^{\mathsf{sess}}$ to determine addresses in the master table $MT$ of the Chameleon scheme and encrypts with the corresponding entries: $C_{\mathsf{CE}} \leftarrow \mathsf{EncCE}(MT, k^{\mathsf{sess}}, M)$. The ciphertext of the fingercasting scheme thus has two elements $C_{\mathsf{BE}}$ and $C_{\mathsf{CE}}$.

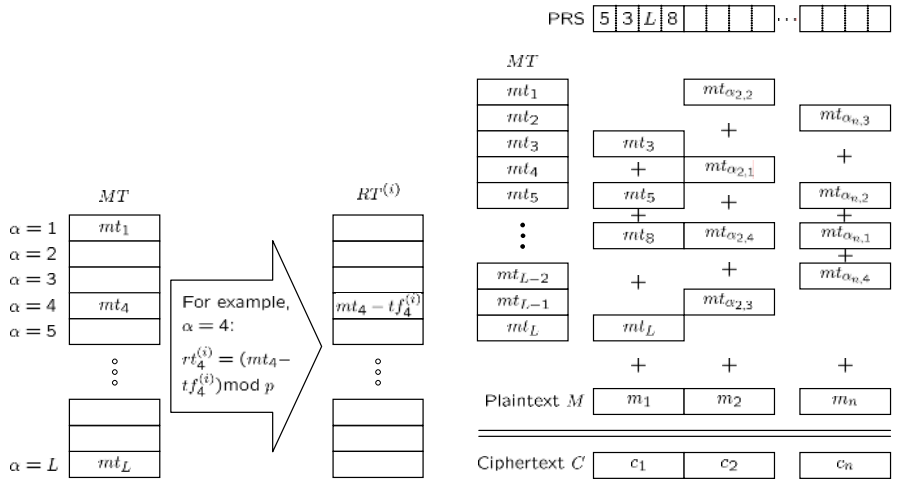
Fourth, the *decryption* algorithm inverts the encryption algorithm with unnoticeable, but characteristic errors. First of all, each non-revoked receiver $u_i$ recovers the correct session key: $k^{\mathsf{sess}} \leftarrow \mathsf{DecBE}(i, SK^{(i)}, C_{\mathsf{BE}})$. Therefore, $u_i$ can recalculate the PRS and the correct addresses in receiver table $RT^{(i)}$. However, this receiver table is slightly different from the master table. Therefore, $u_i$ obtains a fingerprinted copy $M^{(i)}$ that is slightly different from the original content: $M^{(i)} \leftarrow \mathsf{DecCE}(RT^{(i)}, k^{\mathsf{sess}}, C_{\mathsf{CE}})$. Last, the *fingerprint detection* algorithm of the fingercasting scheme is identical to that of the underlying fingerprinting scheme.

### 4.2 A New Chameleon Scheme

Up to now, we have focused on the straightforward aspects of our approach; we have neglected the impact of the requirements on the Chameleon scheme. In the sequel, we will show a specific Chameleon scheme that fulfills all of them. We design it such that its content fingerprints can emulate any additive fingerprinting scheme, which we later instantiate with the SSW scheme as proof of concept.

**Key Generation.** To define this algorithm, we need to determine how the center generates the master table $MT$ and the table fingerprints $TF$. To generate $MT$, the center chooses $L = 2^l$ table entries at random from the interval $[0, z]$ with independent uniform distribution: $mt_\alpha \xleftarrow{R} [0, z]$ for all $\alpha \in \{1, \ldots, L\}$. The center thus obtains the master table $MT := (mt_1, mt_2, \ldots, mt_L)$.

To generate the table fingerprints $TF := (TF^{(1)}, \ldots, TF^{(N)})$, the center selects for each receiver $u_i$ and each master table entry $mt_\alpha$ a fingerprint coefficient in order to disturb the original entry. Specifically, each fingerprint coefficient $tf_\alpha^{(i)}$




**(a)** To derive $RT^{(i)}$ from $MT$, the center subtracts the $L$ fingerprint coefficients $tf_\alpha^{(i)}$ at address $\alpha$ for all $\alpha \in \{1, \ldots, L\}$.

**(b)** To derive $C$ from $M$, the center uses the session key to generate a PRS. It adds the addressed master table entries to $M$.

**Fig. 1.** Receiver table derivation and ciphertext calculation

of table fingerprint $TF^{(i)}$ is independently distributed according to the probability distribution $\mathsf{Prob}$ of the additive fingerprinting scheme, but scaled down with an attenuation factor $f \in \mathbb{R}$, $f \geq 1$:

$$tf_\alpha^{(i)} \leftarrow 1/f \cdot \mathsf{Prob}(par_{\mathsf{FP}}) \tag{1}$$

**Key Extraction.** After the probabilistic key generation algorithm we now describe the deterministic key extraction algorithm. The center processes table fingerprint $TF^{(i)} := (tf_1^{(i)}, \ldots, tf_L^{(i)})$ of receiver $u_i$ as follows: The center subtracts each fingerprint coefficient in $TF^{(i)}$ from the corresponding master table entry to obtain the receiver table entry, which we illustrate in Fig. 1(a):

$$\forall\, \alpha \in \{1, \ldots, L\}: \quad rt_\alpha^{(i)} \leftarrow mt_\alpha - tf_\alpha^{(i)} \bmod p \tag{2}$$

*Remark 1.* The modulo operator allows only integer values to be added. However, $MT$, $TF$, and $M$ are based on real numbers. We solve this ostensible contradiction by scaling the real values to the integer domain with an appropriate scaling factor $\rho$ and ignoring further decimal digits. $\rho$ must be large enough to allow a computation in the integer domain with a sufficiently high precision suitable for the current application. We implicitly assume this scaling to the integer domain whenever real values are used. For example, with real-valued variables $rt^{(i)}$, $mt$, and $tf^{(i)}$ the operation $rt^{(i)} \leftarrow (mt - tf^{(i)}) \bmod p$ actually stands for $\rho \cdot rt^{(i)} \leftarrow (\rho \cdot mt - \rho \cdot tf^{(i)}) \bmod p$. Note that the scaling operation does not give the adversary more information than in the original fingerprinting scheme.

**Encryption.** Fig. 1(b) gives an overview of the encryption algorithm. The session key $k^{\mathsf{sess}}$ is used as the seed of a PRSG with $\mathsf{len}(|k^{\mathsf{sess}}|) \geq n \cdot s \cdot l$, where $s$ will be specified below. To give a practical example for a PRSG, $k^{\mathsf{sess}}$ may serve as the key for a conventional block cipher, e.g., AES, in output feedback mode. Each block of $l$ bits of the PRS is interpreted as an address $\beta$ in the master table $MT$. For each coefficient of the plaintext, the center uses $s$ addresses that define $s$ entries of the master table. In total, the center obtains $n \cdot s$ addresses that we denote with $\beta_{j,k}$, where $j$ is the coefficient index, $k$ the address index, and $\mathsf{Extract}_i$ extracts the $i$-th block of length $l$ from its input string:

$$\forall\, j \in \{1, \ldots, n\},\ \forall\, k \in \{1, \ldots, s\} : \quad \beta_{j,k} \leftarrow \mathsf{Extract}_{(j-1)s+k}(\mathsf{G}(k^{\mathsf{sess}})) \tag{3}$$

For each content coefficient, the center adds the $s$ master table entries modulo the group order. In Fig. 1(b), we illustrate the case $s = 4$, which is the design choice in the original Chameleon cipher. Let $mt_{\beta_{j,k}}$ be the master table entry referenced by address $\beta_{j,k}$ from (3). Then coefficient $c_j$ of $C$ is calculated as

$$c_j \leftarrow \left( m_j + \sum_{k=1}^{s} mt_{\beta_{j,k}} \right) \bmod p\ , \quad j \in \{1, \ldots, n\}\ . \tag{4}$$

**Decryption.** The decryption algorithm proceeds in the same way as the encryption algorithm with two exceptions. First, the receiver has to use its receiver table $RT^{(i)}$ instead of $MT$. Second, the addition is replaced by subtraction. The $j$-th coefficient $m_j^{(i)}$ of the plaintext copy $M^{(i)}$ of receiver $u_i$ is thus calculated as

$$m_j^{(i)} \leftarrow \left( c_j - \sum_{k=1}^{s} rt_{\beta_{j,k}}^{(i)} \right) \bmod p, \tag{5}$$

where $rt_{\beta_{j,k}}^{(i)}$ denotes the receiver table entry of receiver $u_i$ referenced by address $\beta_{j,k}$ generated in (3). As the receiver table $RT^{(i)}$ slightly differs from $MT$, the plaintext copy $M^{(i)}$ obtained by receiver $u_i$ slightly differs from $M$. The ratio of distortion is the same as that of the instantiated fingerprinting scheme.

**Fingerprint Detection.** When the center detects an illegal copy $M^*{=}(m_1^*, \ldots, m_n^*)$ of $M$, it tries to identify the receivers that participated in the generation of $M^*$. To do so, the center verifies whether the fingerprint of a suspect receiver $u_i$ is present in $M^*$. Obviously, the fingerprint is unlikely to appear in its original form; an adversary may have modified it by applying common attacks such as resampling, requantization, and compression. In addition, an adversarial coalition may have colluded and created $M^*$ using several different copies of $M$.

The fingerprint detection algorithm is identical to that of the underlying fingerprinting scheme: $dec \leftarrow \mathsf{DetectFP}(M, M^*, CF^{(i)}, t)$. In order to scale the content fingerprint, we need to select the attenuation factor $f$ in (1). We choose it such that adding $s$ attenuated fingerprint coefficients generates a random variable that follows $\mathsf{Prob}$ *without* attenuation. We give an example in Section 4.3.

In order to verify whether the table fingerprint $TF^{(i)}$ of receiver $u_i$ left traces in $M^*$, $\mathsf{DetectFP}$ calculates the similarity between the detected content fingerprint

$CF^*$ with coefficients $cf_j^* := m_j^* - m_j$ and the content fingerprint $CF^{(i)}$ in $u_i$'s copy $M^{(i)}$ with $cf_j^{(i)} := m_j^{(i)} - m_j \overset{(4),(5)}{=} \sum_{k=1}^{s}(mt_{\beta_{j,k}} - rt_{\beta_{j,k}}^{(i)}) \overset{(2)}{=} \sum_{k=1}^{s} tf_{\beta_{j,k}}^{(i)}$, where $tf_{\beta_{j,k}}^{(i)}$ is the fingerprint coefficient that fingerprinted receiver table $RT^{(i)}$ at address $\alpha = \beta_{j,k}$ in (2). If the similarity is above threshold $t$, the center declares $u_i$ guilty. Note that the calculation of $CF^*$ necessitates the original content $M$, whereas the calculation of $CF^{(i)}$ relies on the session key $k^{\mathsf{sess}}$ and the table fingerprint $TF^{(i)}$; the scheme is thus non-blind. The same algorithm applies to detection of fingerprints in illegal copies of receiver tables. Their fingerprints have the same construction and statistical properties (for further details see [22]).

**Parameter Selection.** The scheme has two major parameters $L$ and $s$ that allow a trade-off between the size of $RT^{(i)}$, which $u_i$ has to store, and the computation cost, which grows linearly with the number $s$ of addresses per content coefficient in (4). By increasing $L$, we can decrease $s$ in order to replace computation cost with storage size. Further details appear in [22].

### 4.3 Instantiation with Spread Spectrum Watermarking

In this section, we instantiate the fingerprinting scheme with the SSW scheme of [14] and thereby inherit its collusion resistance and frame-proofness. Let the center choose the SSW scheme's parameters $par_{\mathsf{FP}} = (\delta, p^{\mathsf{bad}}, p^{\mathsf{pos}})$, which allows to calculate a standard deviation $\sigma'$ and a threshold $t$ via two functions $\mathsf{f}_{\sigma'}(N, n', \delta, p^{\mathsf{bad}})$ and $\mathsf{f}_t(\sigma', N, p^{\mathsf{pos}})$ defined in [14]. The probability distribution of the SSW scheme is then $\mathsf{Prob} = \mathsf{N}(0, \sigma')$. We set $f = s$ such that $1/f \cdot \mathsf{N}(0, \sigma')$ in (1) is still a normal distribution with mean 0 and standard deviation $1/\sqrt{s} \cdot \sigma'$. It remains to define the similarity measure for the detection algorithm $dec \leftarrow \mathsf{DetectFP}(M, M^*, CF^{(i)}, t)$, which [14] defines as $dec = \mathtt{true}$ if $(CF^* \cdot CF^{(i)})/||CF^*|| > t$. We call this instantiation *exact* if it achieves the same statistical properties as the fingerprinting scheme that it instantiates. Theorem 2 below states that the above choice is an exact instantiation; we prove it in [22]:

**Theorem 2.** *Let $\sigma'$ and $\sigma$ be the standard deviations of the SSW scheme and the Chameleon scheme instantiated with SSW, respectively, and $n'$ and $n$ be their number of content coefficients. Then the following mapping between both schemes is an exact instantiation: $\sigma' = \sqrt{s} \cdot \sigma$ $(\Leftrightarrow f = s)$ and $n' = n$*

### 4.4 Analysis

**Correctness, Collusion Resistance and Frame-Proofness.** Correctness follows trivially from the correctness of the two underlying schemes, i.e., the BE scheme and the SSW scheme. Collusion resistance and frame-proofness of content *and* receiver tables follows from the collusion resistance and frame-proofness of the instantiated fingerprinting scheme.

**Security of the Chameleon Encryption Scheme.** In the technical report [22] we reduce the security of our Chameleon scheme $\mathcal{CE}$ to that of the

PRSG with which it is instantiated. To do so, we prove that no efficient algorithm can distinguish the key stream produced by $\mathcal{CE}$ from a truly random key stream.

In the remainder of this section we give the results of the security analysis, whereas further details and the proofs appear in [22]. As pointed out in Remark 1 there is a one-to-one mapping between the actual plaintext symbol space $[0, z]$ containing real numbers with finite precision and the scaled space $\{0, 1, \ldots, Z - 1\}$, which enumerates the elements of $[0, z]$ starting from 0. Further, we define the key symbol space $\mathcal{K}$ to be equal to $\{0, 1, \ldots, Z - 1\}$ and therefore $p := |\mathcal{K}| = Z$. In the sequel, by key symbols we mean the elements of $\mathcal{K}$.

**Definition 1.** *Let $U$ be a uniformly distributed key symbol. Let $p_k$ denote the probability $\mathsf{Pr}\left[X^{(1)} = x_k\right]$ of drawing key symbol $x_k \in \mathcal{K}$ in a single draw $X^{(1)}$ from master table $MT$. Let the* statistical quality *$SQ^{(1)}$ of $MT$ be the statistical difference between $X^{(1)}$ and $U$: $SQ^{(1)} := \frac{1}{2}\sum_{k=0}^{Z-1}\left|p_k - \frac{1}{Z}\right|$. Then we call the master table* strongly converging *if $2SQ^{(1)} \leq d$ for some $d \in \mathbb{R}$ such that $d < 1$.*

**Theorem 3.** *Let $X_k$ denote the $k$-th draw from master table $MT$ and $X^{(s)}$ the random variable resulting from $s$ independent uniformly distributed draws added modulo $Z$: $X^{(s)} := \sum_{k=1}^{s} X_k \bmod Z$. Let $MT$ be a strongly converging master table. Then $X^{(s)}$ has a negligible statistical difference $SQ^{(s)}$ from $U$, where $SQ^{(s)}$ has an upper bound of $\frac{1}{2}d^s$. In addition, no probabilistic polynomial-time adversary can distinguish the key stream of $\mathcal{CE}$ from a truly random key stream.*

**Implementation.** We discuss implementation aspects and practical parameter choices, which allow to trade off storage size for computation cost, in [22].

## 5 Conclusion

In this document we gave a formal proof of the security of a new Chameleon cipher. Applied to a generic fingercasting approach, it provides confidentiality of ciphertext, traceability of content and keys as well as renewability. We achieved confidentiality through a combination of a generic broadcast encryption (BE) scheme and the new Chameleon cipher. In addition, we obtained traceability of keys and content through embedding of a receiver-specific fingerprint into the master table copies, which are given to the receivers. Finally, we achieved renewability through revocation, which is performed in the BE scheme.

## References

1. Touretzky, D.S.: Gallery of CSS descramblers. Webpage, Computer Science Department of Carnegie Mellon University (2000) URL `http://www.cs.cmu.edu/~dst/DeCSS/Gallery` (November 17, 2005).
2. 4C Entity, LLC: CPPM specification—introduction and common cryptographic elements. Specification, Revision 1.0 (2003)
3. AACS Licensing Administrator: Advanced access content system (AACS): Introduction and common cryptographic elements. Specification, Revision 0.90 (2005)

4. Fiat, A., Naor, M.: Broadcast encryption. In Stinson, D.R., ed.: CRYPTO 1993. Volume 773 of LNCS, Springer (1994) 480–491
5. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In Kilian, J., ed.: CRYPTO 2001. Volume 2139 of LNCS, Springer (2001) 41–62
6. Halevy, D., Shamir, A.: The LSD broadcast encryption scheme. In Yung, M., ed.: CRYPTO 2002. Volume 2442 of LNCS, Springer (2002) 47–60
7. Jho, N.S., Hwang, J.Y., Cheon, J.H., Kim, M.H., Lee, D.H., Yoo, E.S.: One-way chain based broadcast encryption schemes. In Cramer, R., ed.: EUROCRYPT 2005. Volume 3494 of LNCS, Springer (2005) 559–574
8. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In Desmedt, Y., ed.: CRYPTO 1994. Volume 839 of LNCS, Springer (1994) 257–270
9. Naor, M., Pinkas, B.: Threshold traitor tracing. In Krawczyk, H., ed.: CRYPTO 1998. Volume 1462 of LNCS, Springer (1998) 502–517
10. Kundur, D., Karthik, K.: Video fingerprinting and encryption principles for digital rights management. Proceedings of the IEEE **92**(6) (2004) 918–932
11. Anderson, R.J., Manifavas, C.: Chameleon—a new kind of stream cipher. In Biham, E., ed.: FSE 1997. Volume 1267 of LNCS, Springer (1997) 107–113
12. Briscoe, B., Fairman, I.: Nark: Receiver-based multicast non-repudiation and key management. In: ACM EC 1999, ACM Press (1999) 22–30
13. Cox, I.J., Kilian, J., Leighton, T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Process. **6**(12) (1997) 1673–1687
14. Kilian, J., Leighton, F.T., Matheson, L.R., Shamoon, T.G., Tarjan, R.E., Zane, F.: Resistance of digital watermarks to collusive attacks. Technical Report TR-585-98, Princeton University, Department of Computer Science (1998)
15. Anderson, R.J., Kuhn, M.: Tamper resistance—a cautionary note. In Tygar, D., ed.: USENIX Electronic Commerce 1996, USENIX (1996) 1–11
16. Maurer, U.: Conditionally-perfect secrecy and a provably-secure randomized cipher. Journal of Cryptology **5**(1) (1992) 53–66
17. Ferguson, N., Schneier, B., Wagner, D.: Security weaknesses in a randomized stream cipher. In Dawson, E., Clark, A., Boyd, C., eds.: ACISP 2000. Volume 1841 of LNCS, Springer (2000) 234–241
18. Ergün, F., Kilian, J., Kumar, R.: A note on the limits of collusion-resistant watermarks. In Stern, J., ed.: EUROCRYPT 1999. Volume 1592 of LNCS, Springer (1999) 140–149
19. Brown, I., Perkins, C., Crowcroft, J.: Watercasting: Distributed watermarking of multicast media. In Rizzo, L., Fdida, S., eds.: Networked Group Communication 1999. Volume 1736 of LNCS, Springer (1999) 286–300
20. Parviainen, R., Parnes, P.: Large scale distributed watermarking of multicast media through encryption. In: Communications and Multimedia Security (CMS 2001). Volume 192 of IFIP Conference Proceedings, Kluwer (2001) 149–158
21. Luh, W., Kundur, D.: New paradigms for effective multicasting and fingerprinting of entertainment media. IEEE Communications Magazine **43**(5) (2005) 77–84
22. Adelsbach, A., Huber, U., Sadeghi, A.R.: Fingercasting—joint fingerprinting and decryption of broadcast messages. Technical Report, Horst Görtz Institute for IT Security (2006) `http://www.prosec.rub.de/publications`.