Unconditionally secure chaffing and winnowing with short authentication tags

D.R. Stinson*
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1, Canada
dstinson@uwaterloo.ca

June 6, 2006

There are just three tasks.
Firstly, I would like to move this pile from here to there,
but I'm afraid that all I have is this tiny tweezers.
Secondly, I would like to empty this well and fill the other;
but I have no bucket, so you'll have to use this eye dropper.
And, lastly, I must have a hole through this cliff,
and here is a needle to dig it.
Norton Juster, *The Phantom Tollbooth.*

Abstract

Rivest proposed the idea of a chaffing-and-winnowing scheme, in which confidentiality is achieved through the use of an authentication code. Thus it would still be possible to have confidential communications even if conventional encryption schemes were outlawed. Hanaoka *et al.* constructed unconditionally secure chaffing-and-winnowing schemes which achieve perfect secrecy in the sense of Shannon. Their schemes are constructed from unconditionally secure authentication codes.

In this paper, we construct unconditionally secure chaffing-and-winnowing schemes from unconditionally secure authentication codes in which the authentication tags are very short. This could be a desirable feature, because certain types of unconditionally secure authentication codes can provide perfect secrecy if the length of an authentication tag is at least as long as the length of the plaintext. The use of such a code might be prohibited if encryption schemes are made illegal, so it is of interest to construct chaffing-and-winnowing schemes based on "short" authentication tags.

*research supported by NSERC discovery grant 203114-06

1 Introduction

The idea of chaffing-and-winnowing was suggested by Rivest [5]. The hypothetical motivating scenario is that encryption schemes might be outlawed at some future time, while the use of message authentication codes (i.e., MACs) could still remain legal. The basic idea of a chaffing-and-winnowing scheme is to use a MAC to provide confidentiality, thus circumventing the hypothetical ban against encryption. Typically, a sender (Alice) and a receiver (Bob) share a secret key L. Alice prepares a large number of authenticated messages, each having the form $m = (x, a)$ where each x is an unencrypted plaintext and a is an authentication tag. Then Alice sends all the authenticated messages to Bob. Bob only accepts the message(s) having authentication tags that are valid under the key L. An observer O has no way to distinguish valid and invalid authentication tags, so O cannot determine the plaintext(s) that Alice is communicating to Bob.

This intriguing idea has not received much study to date. There are two main papers investigating theoretical aspects of chaffing-and-winnowing subsequent to [5], namely Bellare and Boldyreva [1] and Hanaoka *et al.* [4]. The paper [1] gives formal definitions and security treatments of chaffing-and-winnowing schemes, based on the notion of "find-then-guess" security of encryption schemes. The paper [4] studies chaffing-and-winnowing in the setting of unconditional security. The desire is to provide perfect secrecy, based on a suitable unconditionally secure authentication code. That paper also considers "non-malleability" properties of chaffing-and-winnowing schemes. For a paper that discusses practical issues regarding the implementation of chaffing-and-winnowing schemes, see Clayton and Danezis [2].

In this paper, we continue the study of unconditionally secure chaffing-and-winnowing schemes. One possible difficulty with the schemes constructed in [4] is that the underlying authentication codes may already provide perfect secrecy, and thus they might not be considered "legal" authentication codes in our motivating scenario. We are interested in building unconditionally secure chaffing-and-winnowing schemes that are constructed from underlying authentication codes that do not provide perfect secrecy. In fact, we base our construction on authentication codes that employ only one-bit authenticators. Such authenticators clearly cannot provide perfect secrecy for any plaintext space of cardinality greater than 2, so it seems to be an interesting result that we can manufacture unconditionally secure chaffing-and-winnowing schemes from them.

Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{A}$ be a function that computes authentication tags. For a secret key $K \in \mathcal{K}$, $a = f(K, x) \in \mathcal{A}$ is the authentication tag for the plaintext $x \in \mathcal{X}$. All the chaffing-and-winnowing schemes we study in this paper follow the general structure set out in Protocol 1.1. This is essentially the model for unconditionally secure chaffing-and-winnowing introduced in [4].

In view of the structure of the Protocol 1.1, it must be the case that

$$f(L, x') \neq f(L_{x'}, x') \tag{1}$$

for all $x' \neq x$. (If not, then there are at least two authentication tags that are valid under the key L, a contradiction.) This condition (1) is necessary and sufficient for Bob to be able to decrypt m. Notice that (1) implies the weaker condition that $L_{x'} \neq L$ if $x' \neq x$.

Before continuing, we make a few observations and remarks about Protocol 1.1.

Remarks.

1. As is common in the unconditionally secure setting, each key is to be used for only one encryption or decryption.

Protocol 1.1: Unconditionally Secure Chaffing-and-Winnowing Scheme

1. A secret key $L \in \mathcal{K}$ is chosen randomly by Alice and communicated to the receiver, Bob, over a secure channel. L will actually function as a *decryption key*.

2. Later, Alice wants to encrypt a *plaintext* $x \in \mathcal{X}$ to send to Bob. For convenience, suppose that $\mathcal{X} = \{0, \ldots, n-1\}$. Alice chooses an "appropriate" list of keys,

$$\vec{L} = (L_0, \ldots, L_{n-1}),$$

   where $L_x = L$ ($\vec{L}$ is Alice's *encryption key*). Then Alice computes $a_{x'} = f(L_{x'}, x')$ for all x', $0 \leq x' \leq n-1$. The list of n authenticated messages,

$$y = ((0, a_0), \ldots, (n-1, a_{n-1})),$$

   is sent to Bob; y is the *ciphertext*.

3. Bob computes $b_{x'} = f(L, x')$ for all x', $0 \leq x' \leq n-1$. Bob decrypts y to the plaintext x if and only if $\{x' : b_{x'} = a_{x'}\} = \{x\}$. That is, there should be exactly one message $m = (x, a)$ having an authentication tag that is valid under the key L; if so, then this plaintext x is defined to be the decryption of y.

2. In the unconditionally secure setting, we generally consider plaintexts that are chosen from a finite set of possible plaintexts (e.g., bitstrings of some fixed length). In the computationally secure setting, it is more common to consider plaintexts of arbitrary length.

3. In the computationally secure setting, it is not necessary for the ciphertext to include all possible plaintexts. However, this is clearly required if we hope to attain perfect secrecy in the setting of unconditional security. However, in a later section, note that we will consider schemes based on breaking a plaintext into smaller blocks and using multiple keys. This was suggested by Rivest [5] in his original paper, and it can be done in the unconditionally secure setting as well.

4. An interesting feature of chaffing-and-winnowing schemes is that the encryption key is chosen *after* the plaintext to be encrypted is chosen.

5. None of the schemes we construct in this paper are very effficient from the point of view of message expansion, key size, etc. This is because we are attempting to achieve a strong type of confidentiality with a tool that is deliberately chosen in order to *not* provide confidentiality (at least, it does not provide confidentiality when it is used in its intended manner).

1.1 Security of the Scheme

Consider an observer O who sees the ciphertext y in the channel. We do not want O to be able to determine any information about the value of the plaintext x. For every x', $0 \leq x' \leq n-1$, O can

compute

$$\mathcal{L}(x') = \{K \in \mathcal{K} : f(K, x') = a_{x'}\}.$$

$\mathcal{L}(x')$ represents the set of "possible" keys that were used to compute the tag $a_{x'}$. O knows that the correct key, L, is in only one set $\mathcal{L}(x')$ (see item 3 in Protocol 1.1). Therefore O can eliminate from consideration any keys that occur in more than one set $\mathcal{L}(x')$. Therefore, O will compute

$$\mathcal{L}'(x') = \mathcal{L}(x') \setminus \bigcup_{x'' \neq x'} \mathcal{L}(x'')$$

for $0 \leq x' \leq n-1$. Observe that the n sets $\mathcal{L}'(0), \ldots, \mathcal{L}'(n-1)$ are disjoint. O can then deduce that

$$L \in \bigcup_{x'=0}^{n-1} \mathcal{L}'(x').$$

We also observe that (1) can be expressed in the following equivalent form:

$$L_x \in \mathcal{L}'(x). \tag{2}$$

In order to achieve perfect secrecy, it should be the case that every one of the n sets $\mathcal{L}'(x')$ $(0 \leq x' \leq n-1)$ is "equally probable". In particular, if we wish to achieve perfect secrecy, then it is required that

$$\mathcal{L}'(x') \neq \emptyset \text{ for all } x' \in \mathcal{X}. \tag{3}$$

This is because $\mathcal{L}'(x') = \emptyset$ implies that x' cannot be the plaintext, which contradicts the perfect secrecy requirement.

The above-mentioned requirements will usually restrict how Alice chooses keys in step 2 of Protocol 1.1. In a complete specification of a chaffing-and-winnowing scheme, we need to select a suitable authentication code and describe a secure method for Alice to choose encryption and decryption keys.

1.2 Our Contributions

The rest of the paper is organized as follows. In Section 2, we briefly present some background theory about authentication and secrecy codes. In Section 3, we review the main construction from [4] and in Section 4 we give our new construction. We prove our scheme is correct, discuss its efficiency and look at some modifications of it. We also show that our scheme is optimal among chaffing-and-winnowing schemes with one-bit authentication tags. A few final comments are made in Section 5.

2 Unconditionally Secure Authentication and Secrecy

Unconditionally secure authentication codes were first studied by Gilbert, MacWilliams and Sloane [3]. Simmons developed the theoretical foundations (for a survey, see [7]) and many constructions have been provided over the years. We briefly summarize some definitions, notation and basic results.

An *authentication code* is a four-tuple $(\mathcal{X}, \mathcal{A}, \mathcal{K}, f)$ such that $\mathcal{X}$ is a set of *plaintexts*, $\mathcal{A}$ is a set of *authentication tags*, $\mathcal{K}$ is a set of *keys*, and $f : \mathcal{K} \times \mathcal{X} \to \mathcal{A}$. A key $K \in \mathcal{K}$ is chosen uniformly

at random[1] by Alice and communicated to Bob over a secure channel. When Alice wants to send an authenticated message to Bob, she chooses a plaintext $x \in \mathcal{X}$ and computes the authentication tag $a = f(K, x)$. The *message* $m = (x, a)$ is sent to Bob over an insecure channel. Bob *accepts* a message $m = (x, a)$ if and only if $a = f(K, x)$; such a message is said to be *valid* under the key K. In this basic model, each key K is to be used to transmit only one message.

An opponent, denoted by O, may try to cheat Bob by causing him to accept a message that was not constructed and transmitted by Alice. If O introduces a message (x', a') into the channel before Alice sends a message to Bob, then we say that O is attempting to *impersonate* Alice. On the other hand, O might intercept a valid message (x, a) that was transmitted by Alice, and replace it with a message (x', a') where $x' \neq x$. This scenario is termed *substitution*. In each case, O is hoping that Bob will accept the bogus message (x', a').

Let P_I denote the *impersonation probability* of the authentication code, which denotes the maximum probability with which O can fool Bob in an impersonation attack. The following bound on P_I is well-known (see, for example, [8, Theorem 4.14]).

Theorem 2.1. *Suppose that $(\mathcal{X}, \mathcal{A}, \mathcal{K}, f)$ is an authentication code. Then $P_I \geq 1/|\mathcal{A}|$. Further, equality occurs if and only if*

$$|\{K \in \mathcal{K} : f(K, x) = a\}| = \frac{|\mathcal{K}|}{|\mathcal{A}|}$$

for every $x \in \mathcal{X}$ and for every $a \in \mathcal{A}$.

There are also many results about *substitution probabilities* of authentication codes; however, we do not need to consider them in this paper.

Now we turn to unconditionally secure secrecy codes, the theory of which was developed by Shannon [6]. A *secrecy code* is a five-tuple $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, e, d)$ such that $\mathcal{X}$ is a set of *plaintexts*, $\mathcal{Y}$ is a set of *ciphertexts*, $\mathcal{K}$ is a set of *keys*, $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, $e : \mathcal{K} \times \mathcal{Y} \to \mathcal{X}$, and

$$d(K, e(K, x)) = x \tag{4}$$

for all $x \in \mathcal{X}$ and for all $K \in \mathcal{K}$. Observe that (4) implies that $f(K, \cdot)$ is injective for every $K \in \mathcal{K}$.

As before, a key $K \in \mathcal{K}$ is chosen uniformly at random by Alice and communicated to Bob over a secure channel. When Alice wants to send a message to Bob, she chooses a plaintext $x \in \mathcal{X}$ and computes the ciphertext $y = f(K, x)$, which is sent to Bob over an insecure channel. Bob decrypts the ciphertext y to the plaintext $d(K, y)$.

A secrecy code is said to provide *perfect secrecy* if $\mathsf{Pr}[x|y] = \mathsf{Pr}[x]$ for all plaintexts x and all ciphertexts y. That is, the *a priori* probability of plaintext x is the same as the *a posteriori* probability of x given that the ciphertext y is observed. We will assume that $\mathsf{Pr}[x] > 0$ for all x. In this case, we can apply Bayes' Theorem, which states that

$$\mathsf{Pr}[y|x] = \frac{\mathsf{Pr}[x|y] \times \mathsf{Pr}[y]}{\mathsf{Pr}[x]},$$

and it is easily seen that we have perfect secrecy if and only if $\mathsf{Pr}[y|x] = \mathsf{Pr}[y]$ for all plaintexts x and all ciphertexts y.

Shannon gave the following characterization of secrecy codes that provide perfect secrecy.

[1] It is possible to consider codes where keys are not chosen equiprobably. Most of the results we state can be appropriately modified to apply to the more general setting. However, it is simpler and more convenient to restrict our attention to the simplified setting.

Theorem 2.2. *[6] A secrecy code $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, e, d)$ provides perfect secrecy if and only if, for every $y \in \mathcal{Y}$, there exists a non-negative integer r_y such that*

$$|\{K \in \mathcal{K} : e(K, x) = y\}| = r_y$$

for every $x \in \mathcal{X}$.

The following is an immediate corollary of the previous theorem; for a detailed proof, see [8, Theorem 2.4]).

Corollary 2.3. *Suppose that a secrecy code $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, e, d)$ provides perfect secrecy. Then $|\mathcal{K}| \geq |\mathcal{Y}| \geq |\mathcal{X}|$. Further, $|\mathcal{K}| = |\mathcal{Y}| = |\mathcal{X}|$ if and only if, for every $y \in \mathcal{Y}$ and every $x \in \mathcal{X}$, there exists a unique key $K \in \mathcal{K}$ such that $e(K, x) = y$.*

3 The Hanaoka *et al.* Construction

We describe the basic construction from [4]. Define two keys K, K' to be *disjoint* if $f(K, x) \neq f(K', x)$ for all x, $0 \leq x \leq n-1$. In this construction, we require that there exist $|\mathcal{X}| = n$ mutually disjoint keys, which implies that $|\mathcal{A}| \geq n$. The paper [4] suggests to use an optimal authentication code that is secure against impersonation. In view of Theorem 2.1, this is equivalent to saying that $|\mathcal{K}| = |\mathcal{A}| = |\mathcal{M}|$ and the n keys are mutually disjoint.

In this scheme, L is randomly chosen, and then $\vec{L} = (L_0, \ldots, L_{n-1})$ is chosen to be a permutation of $\mathcal{K}$ subject to the constraint that $L_x = L$. It is then clear that

$$|\mathcal{L}(x')| = |\mathcal{L}'(x')| = 1$$

for all x', $0 \leq x' \leq n-1$. From this it is easily seen that the scheme provides perfect secrecy. Here are some specific examples of schemes that can be obtained from the above construction.

Example 3.1. Suppose that $\mathcal{K} = \{0, \ldots, n-1\}$ and $f(K, i) = i - K \bmod n$. This authentication code is optimal (cf. Theorem 2.1). After choosing a plaintext x, Alice defines $L_x = L$, $L_{x+1} = L+1$, $\ldots$, $L_{x-1} = L + n - 1$ (all arithmetic modulo n). Then it is easy to see that the ciphertext is

$$y = ((0, x - L), (1, x - L), \ldots, (n-1, x - L)).$$

Since $a_0 = \cdots = a_{n-1}$ in this scheme, it would be sufficient to simply transmit one copy of this common value, which we denote by a, to Bob. Then, Bob can decrypt the ciphertext by computing $x = a + L \bmod n$. Now, it is easy to see from Corollary 2.3 that the constituent authentication code already provides perfect secrecy. Therefore it could be argued plausibly that this is not a permissible chaffing-and-winnowing scheme.

Example 3.2. Suppose that $\mathcal{K} = \{0, \ldots, n-1\}$ and $f(K, i) = K$ for all i. This is also an optimal authentication code. After choosing a plaintext x, Alice defines $L_x = L$, $L_{x+1} = L + 1$, $\ldots$, $L_{x-1} = L - 1$ (all arithmetic modulo n). Then it is easy to see that the ciphertext is

$$y = (L - x, L - x + 1, \ldots, L - x - 1).$$

In this scheme, Bob can decrypt y to be the unique value x such that $a_x = L$. This authentication code does not provides perfect secrecy. However, it is still the case that $|\mathcal{A}| = |\mathcal{X}| = n$. As mentioned earlier, in a strict prohibition of encryption schemes, it might not be permitted to have $|\mathcal{A}| \geq |\mathcal{X}|$.

These two examples illustrate that the authentication codes used in the construction from [4] may or may not achieve perfect secrecy. We are interested in finding constructions where the underlying authentication codes cannot possibly achieve perfect secrecy.

4 A New Chaffing-and-winnowing Scheme

For our new construction, we use an authentication code in which $\mathcal{A} = \{0,1\}$. For every n-tuple $\kappa = (\kappa_0, \ldots, \kappa_{n-1}) \in \{0,1\}^n$, we define a key K_κ by the rule $f(K_\kappa, i) = \kappa_i$ for $0 \leq i \leq n-1$. Note that there are 2^n keys in $\mathcal{K}$, and each key basically contains a list of authenticators for all n possible plaintexts. For notational convenience, we will also write a key K_κ in the form $K_\kappa = (\kappa_0, \ldots, \kappa_{n-1})$, or even as a string, e.g., $K_\kappa = \kappa_0\kappa_1 \cdots \kappa_{n-1}$.

It is not hard to see that this code has impersonation probability $P_I = 1/2$. This is optimal in that it meets the bound given in Theorem 2.1. Of course, an impersonation probability of $1/2$ does not provide much security.

To illustrate our chaffing-and-winnowing scheme, first we will consider a small example with $n = 4$. Then we will give the general description of the scheme. Suppose that $L = K_\kappa$ where $\kappa = 1011$, and suppose that $x = 2$. Alice could choose the encryption key $\vec{L} = (L_0, L_1, L_2, L_3)$ where

$$\begin{aligned} L_0 &= (0,0,0,1) \\ L_1 &= (1,1,0,1) \\ L_2 &= (1,0,1,1) \\ L_3 &= (1,0,0,0). \end{aligned}$$

For this encryption key, the ciphertext is $y = ((0,0),(1,1),(2,1),(3,0))$. Unambiguous decryption is possible because the list of authenticators, $\vec{a} = (a_0, a_1, a_2, a_3) = (0,1,1,0)$, agrees with the decryption key $L = L_2 = \kappa_0\kappa_1\kappa_2\kappa_3 = 1011$ in a unique co-ordinate. That is, $\vec{a}$ and L have exactly one common co-ordinate, namely, $a_2 = \kappa_2 = 1$. So the decryption of y is $x = 2$, which is correct.

Let's now consider the secrecy of the scheme. For each x', there are eight keys in each $\mathcal{L}_{x'}$:

$$\begin{aligned} \mathcal{L}_0 &= \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111\} \\ \mathcal{L}_1 &= \{0100, 0101, 0110, 0111, 1100, 1101, 1110, 1111\} \\ \mathcal{L}_2 &= \{0010, 0011, 0110, 0111, 1010, 1011, 1110, 1111\} \\ \mathcal{L}_3 &= \{0000, 0010, 0100, 0110, 1000, 1010, 1100, 1110\}. \end{aligned}$$

However, each $\mathcal{L}'_{x'}$ has size 1:

$$\begin{aligned} \mathcal{L}'_0 &= \{0001\} \\ \mathcal{L}'_1 &= \{1101\} \\ \mathcal{L}'_2 &= \{1011\} \\ \mathcal{L}'_3 &= \{1000\}. \end{aligned}$$

In fact, $\mathcal{L}'_{x'} = \{L_{x'}\}$, $x' = 0,1,2,3$. Now, since each $\mathcal{L}'_{x'}$ has size 1, it can be shown that no information about the plaintext can obtained by O upon observation of the ciphertext y (this will be proven more formally a bit later).

The encryption process for our scheme, for arbitrary n, is described in Protocol 4.1. Because condition (1) is satisfied (see Lemma 4.3), it follows that unambiguous decryption is possible.

Protocol 4.1: Encryption in the New Chaffing-and-Winnowing Scheme

1. Suppose that $L = K_\kappa$ is the decryption key, where $\kappa = (\kappa_0, \ldots, \kappa_{n-1})$ and x is the plaintext that Alice wishes to encrypt. Then Alice defines $L_x = L$, and the keys $L_{x'}$ $(x' \neq x)$ are obtained as follows: $L_{x'} = K_{\kappa'}$, where $\kappa' = (\kappa'_0, \ldots, \kappa'_{n-1})$ and

$$\kappa'_j = \begin{cases} \kappa_j + 1 \bmod 2 & \text{if } j = x \text{ or } j = x' \\ \kappa_j & \text{otherwise.} \end{cases}$$

   The encryption key is $\vec{L} = (L_0, \ldots, L_{n-1})$.

2. The ciphertext is $y = ((0, a_0), \ldots, (n-1, a_{n-1}))$ where

$$a_j = \begin{cases} \kappa_j & \text{if } j = x \\ \kappa_j + 1 \bmod 2 & \text{otherwise.} \end{cases}$$

4.1 Perfect Secrecy of the Scheme

In order to show that our scheme provides perfect secrecy, we need to prove that $\mathsf{Pr}[x|y] = \mathsf{Pr}[x]$ for all plaintexts $x \in \{0, \ldots, n-1\}$ and for all ciphertexts $y \in \{0,1\}^n$. Equivalently (as a consequence of Bayes' theorem), we will show that $\mathsf{Pr}[y|x] = \mathsf{Pr}[y]$ for all x and y.

Consider an encryption key, say $\vec{L} = (L_0, \ldots, L_{n-1})$. Define the $n \times n$ array $A_{\vec{L}}$ whose (i,j) entry is $f(L_i, j)$, $0 \leq i, j \leq n-1$. We say that $\vec{L}$ is *admissible* if $A_{\vec{L}}$ has the form

$$A_{\vec{L}} = \begin{pmatrix} a_0 & 1+a_1 & 1+a_2 & \ldots & 1+a_{n-1} \\ 1+a_0 & a_1 & 1+a_2 & \ldots & 1+a_{n-1} \\ 1+a_0 & 1+a_1 & a_2 & \ldots & 1+a_{n-1} \\ 1+a_0 & 1+a_1 & 1+a_2 & \ldots & 1+a_{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1+a_0 & 1+a_1 & 1+a_2 & \ldots & a_{n-1} \end{pmatrix}, \tag{5}$$

where all arithmetic is modulo 2. If $A_{\vec{L}}$ is not admissible, then it is easy to see from Protocol 4.1 that $\mathsf{Pr}[\vec{L}] = 0$.

Given an admissible encryption key $\vec{L}$, the the ciphertext is $y = ((0, a_0), \ldots, (n-1, a_{n-1}))$. Conversely, given y, there is only one admissible encryption key $\vec{L}$ that yields y as the ciphertext, namely, the unique key $\vec{L}$ for which $A_{\vec{L}}$ has the form given above. To summarize, the list of authenticators in y is formed from the diagonal entries of $A_{\vec{L}}$, and every off-diagonal entry of $A_{\vec{L}}$ is the complement of the diagonal entry in the same column. This allows us to define a bijection between the set of ciphertexts y and admissible encryption keys $\vec{L}$.

Let's now compute $\mathsf{Pr}[y|x]$ for fixed y and x. Let $y = ((0, a_0), \ldots, (n-1, a_{n-1}))$ and let $\vec{L} = (L_0, \ldots, L_{n-1})$ be an admissible encryption key. Let $L = L_x$, as usual, and define $\vec{a} = (a_0, \ldots, a_{n-1})$. Then, from (5), it is easy to see that $\mathsf{Pr}[y|L, x] = 0$ unless $\vec{a} + L = e_x + \vec{1} \bmod 2$, where e_x is the n-tuple with a "0" in the xth coordinate and "1" in every other coordinate, and $\vec{1}$ is the vector that has a "1" in every co-ordinate. When $L = e_x + \vec{a} + \vec{1} \bmod 2$, it is easily seen that $\mathsf{Pr}[y|L, x] = 1$.

Since decryption keys are chosen equiprobably, we have that $\mathsf{Pr}[L] = 2^{-n}$ for all L. From these facts, it follows that

$$\mathsf{Pr}[y|x] = \mathsf{Pr}[y|e_x + \vec{a} + \vec{1} \bmod 2, x] \times \mathsf{Pr}[e_x + \vec{a} + \vec{1} \bmod 2] = 2^{-n}.$$

Now we compute

$$\begin{aligned} \mathsf{Pr}[y] &= \sum_{x=0}^{n-1} (\mathsf{Pr}[y|x] \times \mathsf{Pr}[x]) \\ &= \sum_{x=0}^{n-1} (2^{-n} \times \mathsf{Pr}[x]) \\ &= 2^{-n}. \end{aligned}$$

Hence, $\mathsf{Pr}[y|x] = \mathsf{Pr}[y]$ for all y and x, as desired.

4.2 Efficiency of the Scheme

It is possible to prove that the scheme remains unconditionally secure if we restrict the set of decryption keys to be

$$\left\{ K_\kappa : \kappa = (\kappa_0, \ldots, \kappa_{n-1}) \in \{0,1\}^n, \sum_{i=0}^{n-1} \kappa_i = 0 \bmod 2 \right\}.$$

We are reducing the number of decryption keys by a factor of 2 by only using keys with even hamming weight.

The proof that the modified scheme still provides perfect secrecy is almost identical to the previous proof. It is important to check that every row of A will have even hamming weight, the vector $\vec{a}$ has hamming weight congruent to $n+1$ modulo 2, and the hamming weight of $e_x + \vec{a} + \vec{1}$ is even. Then the previous proof can be modified by replacing all occurrences of 2^{-n} by 2^{-n+1} in the computations of the probabilities $\mathsf{Pr}[y|x]$ and $\mathsf{Pr}[y]$.

We will measure the efficiency of the chaffing-and-winnowing scheme in terms of the length of ciphertext and the length of the key (in bits). Suppose that $|\mathcal{X}| = n = 2^k$. Then there are 2^{n-1} possible decryption keys in the improved scheme, so the length of a decryption key is $n-1 = 2^k - 1$ bits. A ciphertext has the form $y = ((0, a_0), \ldots, (n-1, a_{n-1}))$. Since the first co-ordinates are $0, \ldots, n-1$, in that order, we can eliminate them from the ciphertext if desired, and just transmit the vector of authenticators, $\vec{a} = (a_0, \ldots, a_{n-1})$. Under this assumption, the length of the ciphertext is $n = 2^k$.

For reference, we will denote the above-described scheme by $\mathsf{CW}(k)$. Summarizing, we have shown the following.

Theorem 4.1. *For any integer $k \geq 1$, the scheme $\mathsf{CW}(k)$ is an unconditionally secure chaffing-and-winnowing scheme for k-bit plaintexts, based on 1-bit authenticators, in which an encryption key consists of $2^k - 1$ bits and a ciphertext consists of 2^k bits.*

4.3 An Efficiency Improvement

Suppose we have an ℓ-bit plaintext, where $\ell = rk$, and we break it into r blocks, each of which contains k bits. Each k-bit block is then encrypted using a scheme $\mathsf{CW}(k)$. In total, we have r independent schemes $\mathsf{CW}(k)$, each of which has an independently chosen key. Each possible ℓ-bit plaintext receives an r-bit authenticator, which is the concatenation of the 1-bit authenticators of each of the r blocks in the plaintext. This hybrid scheme, which will be denoted by $\mathsf{HCW}(r, k)$, has the following properties.

Theorem 4.2. *For integers $k, r \geq 1$, the scheme $\mathsf{HCW}(r, k)$ is an unconditionally secure chaffing-and-winnowing scheme for rk-bit plaintexts, based on r-bit authenticators, in which an encryption key consists of $r(2^k - 1)$ bits and a ciphertext consists of $r\,2^k$ bits.*

Theorem 4.2 illustrates a trade-off between authenticator size and efficiency. The parameter k denotes the ratio between the length of a plaintext and the length of an authenticator in the scheme $\mathsf{HCW}(r, k)$. We have previously suggested that the situation when $k = 1$ is problematic because the underlying authentication schemes might already provide perfect secrecy. Clearly, as k is increased, the system becomes less and less efficient, so the "best value" of k to use is the smallest one that would be permitted by law.[2]

For example, when we set $k = 1$, our scheme $\mathsf{HCW}(r, 1)$ has r-bit plaintexts and r-bit authenticators, an r-bit encryption key and a $2r$-bit ciphertext. This is a two-fold message expansion, as compared to the classical one-time pad or any other optimal secrecy code that provides perfect secrecy (see Corollary 2.3). For the next case, $k = 2$, we have $2r$-bit plaintexts and r-bit authenticators, a $3r$-bit encryption key and a $4r$-bit ciphertext. For larger values of k, the situation becomes progressively worse.

4.4 Optimality of the Schemes $\mathsf{CW}(k)$

It is in fact possible to show that the schemes $\mathsf{CW}(k)$ are optimal in the set of all chaffing-and-winnowing schemes that have 1-bit authenticators and k-bit plaintexts. To show this, we sketch a proof that there must be at least 2^{n-1} keys in such a scheme, where $n = 2^k$.

The optimality proof involves showing that the method of constructing keys that we used in Protocol 4.1 is the only way to attain perfect secrecy. First, we prove a preliminary lemma.

Lemma 4.3. *Suppose that $L = K_\kappa$ and let x be the plaintext. Then $\vec{L}$, as described in Protocol 4.1, is the only encryption key such that (2) and (3) are both satisfied.*

Proof. First, if condition (2) holds, then

$$f(L_{x'}, x') = f(L, x') + 1 = \kappa_{x'} + 1$$

for all $x' \neq x$. This determines the values of the diagonal elements of the matrix $A_{\vec{L}}$, which we denote by $a_0, a_1, \ldots, a_{n-1}$, respectively, as in (5).

Now, let $x' \neq x$. We require that $\mathcal{L}(x') \neq \emptyset$. Suppose that $L^* = K_{\kappa^*} \in \mathcal{L}(x')$, where $\kappa^* = (\kappa_0^*, \ldots, \kappa_{n-1}^*)$. We have that $L^* \in \mathcal{L}(x')$ and $L^* \notin \mathcal{L}(j)$ for all $j \neq x'$. Hence, it follows that

$$\kappa_j^* = \begin{cases} a_j & \text{if } j = x' \\ 1 + a_j \bmod 2 & \text{otherwise.} \end{cases}$$

[2] Of course, encryption is currently legal, so this discussion is purely hypothetical.

Therefore, $A_{\vec{L}}$ has the form given in (5) and the desired result is proven. □

Now, let L be any decryption key. From Lemma 4.3, letting x vary over all n possible plaintexts, it follows that every binary n-tuple that is hamming distance 2 from L is also a decryption key. From this, it is easily seen that $\mathcal{K}$ contains all the binary n-tuples that are even hamming distance from L, so $|\mathcal{K}| \geq 2^{n-1}$. In fact, $\mathcal{K}$ must consist of all the binary n-tuples of even weight, all the binary n-tuples of odd weight, or all the binary n-tuples.

5 Conclusion

We do not claim that our scheme is "practical". The interesting contribution of our paper is that we can still obtain perfect secrecy even when we are forced to use particularly ill-suited tools, namely, authentication codes with very short authentication tags.

It would be of interest to develop bounds (i.e., necessary conditions) on the parameters of unconditionally secure chaffing-and-winnowing schemes. We have shown that our schemes $\mathsf{CW}(k)$ are optimal in the set of all chaffing-and-winnowing schemes that have 1-bit authenticators and k-bit plaintexts. We ask if there are more efficient "r-bit schemes" than the schemes $\mathsf{HCW}(r, k)$ that we have constructed.

References

[1] M. Bellare and A. Boldyreva. The security of chaffing and winnowing. *Lecture Notes in Computer Science* **1976** (2000), 517–530 (ASIACRYPT 2000).

[2] R. Clayton and G. Danezis. Chaffinch: confidentiality in the face of legal threats. *Lecture Notes in Computer Science* **2578** (2003), 70–86 (Information Hiding 2002).

[3] E.N. Gilbert, F.J. MacWilliams and N.J.A. Sloane. Codes which detect deception. *Bell System Tech. J.* **53** (1974), 405–424.

[4] G. Hanaoka, Y. Hanaoka, M. Hagiwara, H. Watanabe and H. Imai. Unconditionally secure chaffing-and-winnowing: a relationship between encryption and authentication. *Lecture Notes in Computer Science* **3857** (2006), 154–162 (AAECC-16).

[5] R.L. Rivest. Chaffing and winnowing: confidentiality without encryption. *CryptoBytes* **4-1** (1998), 12–17.

[6] C.E. Shannon. Communication theory of secrecy systems. *Bell System Tech. J.* **28** (1949), 656–715.

[7] G.J. Simmons. A survey of information authentication. In "Contemporary cryptology", IEEE Press, New York, 1992, pp. 379–419.

[8] D.R. Stinson. *Cryptography Theory and Practice, Third Edition*, Chapman & Hall/CRC, 2006.