Unconditionally Secure Key Agreement Protocol

Cyril Prissette

Signal - Information - Systèmes
Université de Toulon et du Var
83130 La Garde, France prissette@univ-tln.fr

Abstract. The key agreement protocol are either based on some computational infeasability, such as the calculus of the discrete logarithm in [1], or on theoretical impossibility under the assumption that Alice and Bob own specific devices such as quantum channel [2]. In this article, we propose a new key agreement protocol called CHIMERA which requires no specific device. This protocol is based on a generalization we propose of the reconciliation algorithm. This protocol is proved unconditionally secure.

1 Introduction

The security of cryptographic systems is based either on a computational infeasability or an a theoretical impossibility. However, some cryptographic problems have no known unconditionally secure solution. For example, the key agreement problem has computational secure solutions, as the Diffie-Hellman protocol [1], but no unconditional secure solution under the assumption that Alice and Bob has no specific equipment such as quantum channel, deep-space radio source, or satellite.

Our work is inspired by these protocols and uses a generalized version of an interactive error-correcting algorithm proposed by C.H. Bennett and G. Brassard in [2]. This algorithm, called reconciliation, fits the parameter of the quantum channel, but is insecure for our protocol because of some properties of the sequences we use. The first part of this paper is a presentation of the generalization of the reconciliation algorithm.

The next part is a presentation of CHIMERA, which is a key agreement protocol with unconditional security. It uses information-theoretic algorithms such as generalized reconciliation and extended Huffman coding.

In [3], U. Maurer gives a general description of key agreement protocols and the conditions a key agreement protocol must satisfy to be secure [4],[5]. We recall these conditions and prove that CHIMERA satisfy all this conditions if the value of a parameter of the protocol is in a given range. Next, we propose a particular value of this parameter in the given range to optimize the length of the key created by CHIMERA.

B. Honary (Ed.): Cryptography and Coding 2001, LNCS 2260, pp. 277–293, 2001.

2 Generalized Reconciliation

2.1 Bennett and Brassard's Reconciliation

The reconciliation process is, as describe in [2], an iterative algorithm which destroy errors between two binary sequences A and B owned by Alice and Bob. The destruction of the errors is secure even if Eve listen the insecure channel used by Alice and Bob to perform reconciliation. The algorithm does not destroy all errors between the two sequences in one round, but it can be repeated several times to destroy statistically all the errors. The price to pay to obtain to identical sequence is the sacrifice of bits of the sequences and thus, the reduction of the length of the sequences.

Here is the algorithmic description of one round of reconciliation :

Alice and Bob cut their sequences A and B into subsequences of length k. For each sub-sequence $(A_i, \ldots, A_{i+k-1})$ from A and $(B_i, \ldots, B_{i+k-1})$ from B, they send each other (on the public insecure channel) the parity of their sub-sequence.

- If the parity of the sub-sequence $(A_i, \ldots, A_{i+k-1})$ differs from the parity of the sub-sequence $(B_i, \ldots, B_{i+k-1})$, Alice and Bob destroy their respective sub-sequences.
- Else Alice and Bob destroy respectively A_{i+k-1} and B_{i+k-1}, and keep $(A_i, \ldots, A_{i+k-2})$ and $(B_i, \ldots, B_{i+k-2})$.

The principle is simple : if the parities differ, then the sub-sequences differ. if Alice and bob destroy these sub-sequences, they destroy (at least) one error between the two sequences.

On the other hand, if the parities are equal. This does not mean that the two sequences are equal. However Eve knows one bit of information about the subsequence : so, Alice and Bob destroy one bit from their subsequence.

Obviously, the reconciliation works only if the sequences A abd b are close enough, and is secure only if Eve has no information about A and B before the reconciliation. For example, if she knows with certainty the value of one bit from A and B and if Alice and Bob use sub-sequences of length two, she learns from the parities of the sequences the whole sequences and so the bit kept if the parity are equals.

2.2 Generalized Reconciliation

Sometimes, in particular in CHIMERA, the parity of a sub-sequence reveals more information than the entropy of one bit of the subsequence. This happens, for example, when $p(A_i = 0) < p(A_i = 1)$.

The generalized reconciliation algorithm REC(k,n), which is as follows, let Alice and Bob sacrifice n symbols (instead of only one) of their sub-sequences of length k when the parities are equals.

Alice and Bob cut their sequences A and B into subsequences of length k. For each sub-sequence $(A_i, \ldots, A_{i+k-1})$ from A and $(B_i, \ldots, B_{i+k-1})$ from B, they send each other (on the public insecure channel) the parity of their sub-sequence.

- If the parity of the sub-sequence $(A_i, \ldots, A_{i+k-1})$ differs from the parity of the sub-sequence $(B_i, \ldots, B_{i+k-1})$, Alice and Bob destroy their respective sub-sequences.
- Else Alice and Bob destroy respectively A_{i+k-n} and B_{i+k-n}, and keep $(A_i, \ldots, A_{i+k-n-1})$ and $(B_i, \ldots, B_{i+k-n-1})$.

The principle is the same than in Bennett and Brassard reconciliation R(k,1) : if the parities differs,then the sub-sequence contain errors, so Alice and Bob destroy the sub-sequences. Otherwise, Alice and Bob destroy more information than the information revealed by the parities.

The generalization of the reconciliation algorithm is very useful in our protocol, called CHIMERA, which uses REC(3,2). Actually, in this protocol the sequences are biased but the entropy of two bits is always greater than the entropy of the parity of three bits. This property is proved in the section (7).

3 Presentation of CHIMERA

The CHIMERA is a key agreement protocol. we present it with some parameters which are optimal and insure its security. The choice of the values used in CHIMERA is explain in the study of the protocol which follows this presentation.

The following protocol allows Alice and Bob to build a secret common quantity of length 128 bits.

- Alice builds a binary sequence $A^{[0]}$ with the following properties :
  - $|A^{[0]}| = 2000000$
  - $\forall_i \; p(A_i^{[0]} = 1) = p_b = \frac{3}{16}$
- Bob builds a binary sequence $B^{[0]}$ with the following properties :
  - $|B^{[0]}| = 2000000$
  - $\forall_i \; p(B_i^{[0]} = 1) = p_b = \frac{3}{16}$
- Alice and Bob repeat 6 times the following reconciliation algorithm REC(3,2) on their respective sequences (We note $A^{[k]}$ and $B^{[k]}$ Alice and Bob's sequences after k rounds of reconciliation).

```
l=0
forall $i$ such as ($i < |A^{[k]}| - 2$ and $i \bmod 3 = 0$)
        if ($\bigoplus_{j=0}^{2} A_{i+j}^{[k]}) = \bigoplus_{j=0}^{2} B_{i+j}^{[k]})$) then
                $A_l^{[k+1]} \leftarrow A_i^{[k]}$
                $B_l^{[k+1]} \leftarrow B_i^{[k]}$
                $l \leftarrow l + 1$
        end if
end forall
```

- Alice compresses the sequence $A^{[6]}$ with the extended Huffman code $\mathcal{H}$ using 11-tuples as symbols of the language. The resulting sequence is the key S.

- Bob compresses the sequence $B^{[6]}$ with the extended Huffman code $\mathcal{H}$ using 11-tuples as symbols of the language. The resulting sequence is the key S'.

Alice and Bob have the same quantity $S = S'$ of length 128.

4 Properties of Key Agreement Protocols

In [3], U. Maurer gives the properties a key agreement have to satisfy. These properties come from [4] and [5]. They are conditions of soundness and security.

Considering that Eve is passive, a key agreement protocol which creates binary sequences S and S' by exchanging between Alice and Bob t messages $C_1, \ldots, C_t$ must satisfy the three conditions

- $P[S \neq S'] \approx 0$: Alice and bob must obtain with a very high probability the same sequence.
- $H(S) \approx |S|$: the key must be very close to uniformly distributed.
- $I(S; C^t Z) \approx 0$: Eve has no information about S, considering her initial knowledge Z and her eavesdropping of the insecure channel.

Moreover, the goal of the key-agreement is to make the length of the key S as long as possible.

The CHIMERA satisfied each of these properties. The proof that each property is satisfied is given in the three following sections of this paper. For each proof, we assume that the bias p_b of the initial sequences $A^{[0]}$ and $B^{[0]}$ is in the range $[0 : \frac{1}{2})$, and we search the conditions on this parameter the CHIMERA have to respect to work and be sure. We also assume the reconciliation needs r round to create identical sequences and the extended Huffman code uses n-tuples.

Then, under the conditions on p_b obtained in each proof we explain the choice of the values $p_b = \frac{3}{16}$, $r = 6$ and $n = 11$.

5 Proof of the Property $P[S \neq S'] \approx 0$

The proof of the property $P[S \neq S'] \approx 0$ is based on the study of the distance evolution between Alice's sequence $A^{[i]}$ and Bob's sequence $B^{[i]}$ after i rounds of reconciliation.

5.1 Definition : Normalized Distance

The normalized distance $d_N(A, B)$ between to sequences of bits A and B is defined as the ration between Hamming distance $d_H(A, B)$ and the length $|A|$ of the sequences.

$$d_N(A, B) = \frac{d_H(A, B)}{|A|}. \tag{1}$$

5.2 Initial Normalized Distance $d_N(A^{[0]}, B^{[0]})$

Let p_b be the biased probability of the random generators. The initial normalized distance is a function of p_b. The following table presents the four possible values of the couple $(A_i^{[0]}, B_i^{[0]})$ with their occurrence probability.

Table 1. Possible values of $(A_i^{[0]}, B_i^{[0]})$ with occurence probability.

$A_i^{[0]}$	$B_i^{[0]}$	$p(A_i^{[0]}, B_i^{[0]})$
0	0	$(1-p_b)^2$
1	1	p_b^2
0	1	$p_b(1-p_b)$
1	0	$p_b(1-p_b)$

In the two last cases $A_i^{[0]}$ and $B_i^{[0]}$ differs, so $p(A_i^{[0]} \neq B_i^{[0]}) = 2p_b(1-p_b)$. This result can be extended to the whole sequences to obtain the average Hamming distance $d_H(A^{[0]}, B^{[0]}) = |A^{[0]}|2p_b(1-p_b)$. So the initial normalized distance between $A^{[0]}$ and $B^{[0]}$ is :

$$d_N(A^{[0]}, B^{[0]}) = \frac{d_H(A^{[0]}, B^{[0]})}{|A^{[0]}|} = 2p_b(1-p_b). \tag{2}$$

In CHIMERA, we set $p_b \in [0 : \frac{1}{2})$. So we have the following range for the initial normalized distance between S and S' which is a function of the bias of the random generators used to build $A^{[0]}$ and $B^{[0]}$:

$$d_N(A^{[0]}, B^{[0]}) \in [0 : \frac{1}{2}). \tag{3}$$

5.3 Evolution of the Normalized Distance $d_N(A^{[k]}, B^{[k]})$

Let $d_N(A^{[k]}, B^{[k]})$ be the normalized distance between $A^{[k]}$ and $B^{[k]}$ after k rounds of reconciliation with the algorithm $REC(3,2)$. The following table presents the 32 possible values of the two 3-tuples $(A_i^{[k]}, A_{i+1}^{[k]}, A_{i+2}^{[k]})$ and $(B_i^{[k]}, B_{i+1}^{[k]}, B_{i+2}^{[k]})$ with their occurrence probability when the bits $A_i^{[k]}$ and $B_i^{[k]}$ are kept (i is a multiple of 3).

The 16 first cases give $A_i^{[k]} = B_i^{[k]}$, which means that the reconciliation $REC3$ works and the distance reduces. At the opposite, the 16 last cases gives $A_i^{[k]} \neq B_i^{[k]}$, the reconciliation REC(3,2) fails and the distance increases.

The normalized distance $d_N(A^{[k+1]}, B^{[k+1]})$ after one more round of reconciliation REC(3,2) is a function of $d_N(A^{[k]}, B^{[k]})$. It is given by the ratio between the probability of the 16 last cases and the probability of the 32 cases (we set $d_N = d_N(A^{[k]}, B^{[k]})$) :

$$d_N(A^{[k+1]}, B^{[k+1]}) = \frac{2(1-d_N)d_N^2}{3(1-d_N)d_N^2 + (1-d_N)^3}. \tag{4}$$

Table 2. Possibles values of $(A_i^{[k]}, A_{i+1}^{[k]}, A_{i+2}^{[k]}, B_i^{[k]}, B_{i+1}^{[k]}, B_{i+2}^{[k]})$ with occurrence probability

$A_i^{[k]}$	$A_{i+1}^{[k]}$	$A_{i+2}^{[k]}$	$B_i^{[k]}$	$B_{i+1}^{[k]}$	$B_{i+2}^{[k]}$	$p(A_i^{[k]}, A_{i+1}^{[k]}, A_{i+2}^{[k]}, B_i^{[k]}, B_{i+1}^{[k]}, B_{i+2}^{[k]})$
0	0	0	0	0	0	$(1 - d_N(A^{[k]}, B^{[k]}))^3$
0	0	0	0	1	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
0	0	1	0	0	1	$(1 - d_N(A^{[k]}, B^{[k]}))^3$
0	0	1	0	1	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
0	1	0	0	0	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
0	1	0	0	1	0	$(1 - d_N(A^{[k]}, B^{[k]}))^3$
0	1	1	0	0	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
0	1	1	0	1	1	$(1 - d_N(A^{[k]}, B^{[k]}))^3$
1	0	0	1	0	0	$(1 - d_N(A^{[k]}, B^{[k]}))^3$
1	0	0	1	1	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	0	1	1	0	1	$(1 - d_N(A^{[k]}, B^{[k]}))^3$
1	0	1	1	1	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	1	0	1	0	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	1	0	1	1	0	$(1 - d_N(A^{[k]}, B^{[k]}))^3$
1	1	1	1	0	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	1	1	1	1	1	$(1 - d_N(A^{[k]}, B^{[k]}))^3$
0	0	0	1	0	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
0	0	0	1	1	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
0	0	1	1	0	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
0	0	1	1	1	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
0	1	0	1	0	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
0	1	0	1	1	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
0	1	1	1	0	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
0	1	1	1	1	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	0	0	0	0	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	0	0	0	1	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	0	1	0	0	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	0	1	0	1	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	1	0	0	0	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	1	0	0	1	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	1	1	0	0	1	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$
1	1	1	0	1	0	$(1 - d_N(A^{[k]}, B^{[k]}))(d_N(A^{[k]}, B^{[k]}))^2$

5.4 Limit of the Normalized Distance $d_N(A^{[k]}, B^{[k]})$

Proving that $\forall d_N(A^{[0]}, B^{[0]}) \in [0 : \frac{1}{2}), \lim_{r \to +\infty} d_N(A^{[k]}, B^{[k]}) = 0$ is equivalent to prove $P[S \neq S'] \approx 0$. We do not consider the last computation of the protocol (the Huffman coding of the sequences S and S') because Alice and Bob obtain the same sequence after this compression if they have the same sequence before this compression. So we only have to prove the normalized distance between $A^{[r]}$

and $B^{[r]}$ to be equal to zero before the Huffman coding, i.e after the reconciliation rounds.

The limits of $d_N(A^{[k]}, B^{[k]})$ are the roots of the equation

$$d = \frac{2d^2}{3(1-d)^2 + (1-d)^3}. \tag{5}$$

This equation can be re-write as :

$$d(1-d)(d-\frac{1}{2})^2 = 0. \tag{6}$$

Obviously, the roots of this equation,and so the possible limits of the normalize distance between $A^{[k]}$ and $B^{[k]}$ after k rounds of reconciliation, are $\{0, \frac{1}{2}, 1\}$.

$$\lim_{k \to +\infty} d_N^{[k]}(S, S') \in \{0, \frac{1}{2}, 1\}. \tag{7}$$

Let us consider now the case $d_N(A^{[0]}, B^{[0]}) \in [0 : \frac{1}{2})$ seen in (3) which is encountered in CHIMERA and study the limit of the normalized distance between $A^{[k]}$ and $B^{[k]}$ for this initial range of value. In this range, the next inequality is true :

$$\forall d_N^{[0]}(S, S') \in [0 : \frac{1}{2}), \frac{2d^2}{3(1-d)^2 + (1-d)^3} < d. \tag{8}$$

So, re-writing the equation with the normalized distance evolution function (4), we have:

$$\forall d_N^{[0]}(S, S') \in [0 : \frac{1}{2}), d_N(A^{[k+1]}, B^{[k+1]}) < d_N(A^{[k]}, B^{[k]}). \tag{9}$$

For $d_N(A^{[0]}, B^{[0]}) \in [0 : \frac{1}{2})$, the sequence $\{d_N(A^{[k+1]}, B^{[k+1]})\}_{k \geq 0}$ is decreasing and bounded. So it is convergent and its limit is 0.

$$\forall d_N(A^{[0]}, B^{[0]}) \in [0 : \frac{1}{2}), \lim_{k \to +\infty} d_N(A^{[k]}, B^{[k]}) = 0. \tag{10}$$

So after enough rounds, noted r, of reconciliation the normalized distance between $A^{[k]}$ and $B^{[k]}$ becomes as close to zero as wanted. This means that the sequences are equal, with a very high probability.

$$\forall d_N(A^{[0]}, B^{[0]}) \in [0 : \frac{1}{2}), \forall \epsilon > 0, \exists r, d_N(A^{[r]}, B^{[r]}) < \epsilon. \tag{11}$$

Choosing ϵ very close to 0, we can write :

$$P[A^{[r]} = B^{[r]}] \approx 0. \tag{12}$$

Obviously, the Huffman coding $\mathcal{H}$ does not change this result. We note $\mathcal{H}(A^{[r]})$ and $\mathcal{H}(B^{[r]})$, the Huffman coding of $A^{[r]}$ and $B^{[r]}$ respectively. So,

$$P[\mathcal{H}(A^{[r]}) = \mathcal{H}(B^{[r]})] \approx 0. \tag{13}$$

As defined in CHIMERA, the sequences $\mathcal{H}(A^{[r]})$ and $\mathcal{H}(B^{[r]})$ are the keys and can be noted, in accordance with [3], S and S'. So, we have :

$$P[S \neq S'] \approx 0. \tag{14}$$

6 Proof of the Property $|S| \approx H(S)$

The proof of the property $|S| \approx H(S)$ is based on the evaluation of the normalized weight of the sequences $A^{[r]}$ and $B^{[r]}$ and on a property of the Huffman code.

6.1 Definition : Normalized Weight

The normalized weight $\omega_N(A)$ of the binary sequence A is defined as the ratio between Hamming weight $\omega_H(A)$ and the length $|A|$.

$$\omega_N(A) = \frac{\omega_H(A)}{|A|}. \tag{15}$$

Of course, the initial normalized weight of the sequences $A^{[0]}$ and $B^{[0]}$ is equal to p_b.

6.2 Residual Normalized Weight

We consider the residual normalized weight of the sequences $A^{[r]}$ and $B^{[r]}$, i.e. when the condition $(P[S \neq S'] = 0)$ is satisfied. We note p_k the probability of keeping a bit after r rounds of reconciliation. This probability, we will not evaluate now, is function of the number of reconciliation rounds (each round divide by three, at least, the length of the sequences) and of the normalized distance of the sequences for each round of reconciliation (the closest the sequences are, the highest is the probability to keep a given bit).

As we keep only identical bits and sacrifice a certain amount of bits for security, the following table presents the two values the ith bit of $A^{[r]}$ and $B^{[r]}$ can have, with the probability associated to each case.

Table 3. Possibles values of $A_i^{[r]}$ and $B_i^{[r]}$ with occurrence probability

$A_i^{[r]}$	$B_i^{[r]}$	$p(A_i^{[r]}, B_i^{[r]})$
0	0	$(1-p_b)^2 p_k$
1	1	$p_b^2 p_k$

Obviously, the normalized weight of $A^{[r]}$ (and $B^{[r]}$) at the end of the reconciliation is :

$$\omega_N(A^{[r]}) = \frac{p_b^2 p_k}{(1-p_b)^2 p_k + p_b^2 p_k} = \frac{p_b^2}{(1-p_b)^2 + p_b^2}. \tag{16}$$

This result is validated by simulations as one can see in the following graph representing $\omega_N(A^{[r]})$ as a function of p_b :

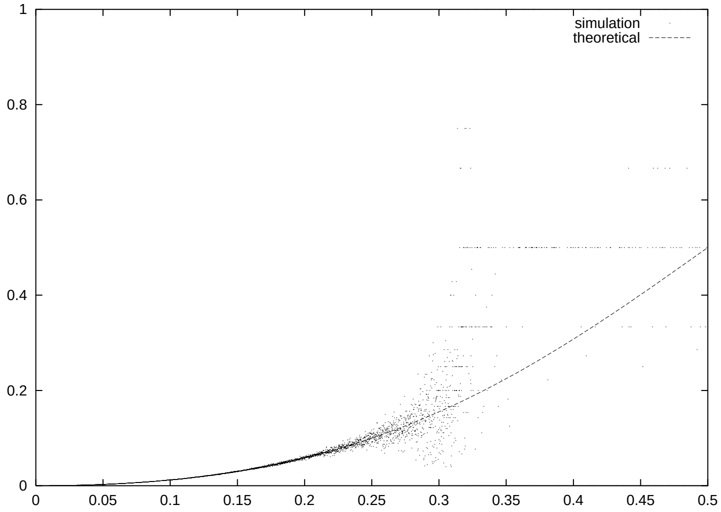

Fig. 1. This graph shows $\omega_N(A^{[r]}$ as a function of p_b. The curve is given by the theory. The dots are simulation results

Note that for $p_b > \frac{1}{4}$, the simulation results are noisy because the residual length of the sequence becomes too small. So, we will avoid this range of value for the bias of the random generators used to build Alice and Bob's sequences.

6.3 Entropy of $\mathcal{H}(A^{[r]})$

As $\omega_N(A^{[r]}) < \frac{1}{2}$, the entropy of $A^{[r]}$ is not maximal [6]. However, the last stage of the protocol is the compression of the sequences with an extended Huffman code. It is well known that using big t-tuples as the symbols of the language improves the compression ratio. With big enough t-tuples, the compression ratio is near of the entropy of the sequence. Noting $\mathcal{H}$, the extended Huffman code, we have :

$$|\mathcal{H}(A^{[r]})| \approx H(\mathcal{H}(A^{[r]})). \tag{17}$$

As $\mathcal{H}(A^{[r]})$ is the sequence S, we can rewrite the preceding equation :

$$H(S) \approx |S|. \tag{18}$$

7 Proof of the Property $I(S; C^t Z) \approx 0$

The proof of the property $I(S; C^t Z) \approx 0$ is based on the comparison of the amount of information revealed and sacrificed by the reconciliation algorithm. We will only study the cases in which bits are kept : when the bits are destroyed because they are different, the information that Eve can gather is useless.

Moreover, as Eve has no information about $A^{[0]}$ and $B^{[0]}$, we can forget Z and just prove that

$$I(S; C^t) \approx 0. \tag{19}$$

7.1 Information Sacrificed by the Reconciliation

Let us consider the reconciliation of the 3-tuples $(A_i^{[k]}, A_{i+1}^{[k]}, A_{i+2}^{[k]})$ from Alice's sequence and $(B_i^{[k]}, B_{i+1}^{[k]}, B_{i+2}^{[k]})$ from Bob's sequence (i is a multiple of 3). When for a given 3-tuples one bit is kept, then 2 bits are destroyed. Moreover, the sacrificed bits are independent from each other. So, the amount of information sacrificed is

$$H_s = 2H(\omega_N(A^{[k]}). \tag{20}$$

7.2 Information Revealed by the Reconciliation

Now, let us consider the information revealed by the reconciliation, i.e. the parity of the 3-tuple $(A_i^{[k]}, A_{i+1}^{[k]}, A_{i+2}^{[k]})$:

$$H(C_i^{2k+1}) = H(\bigoplus_{j=0}^{2} A_{i+j}^{[k]}). \tag{21}$$

The following table gives the probability of incidence of each case :

Table 4. Possible values of $(A_i^{[k]}, A_{i+1}^{[k]}, A_{i+2}^{[k]})$ with occurence probability.

$A_i^{[k]}$	$A_{i+1}^{[k]}$	$A_{i+2}^{[k]})$	$\bigoplus_{j=0}^{2} A_{i+j}^{[k]}$	$p(A_i^{[k]}, A_{i+1}^{[k]}, A_{i+2}^{[k]})$
0	0	0	0	$(1-\omega_N(A^{[k]}))^3$
0	1	1	0	$(1-\omega_N(A^{[k]}))\omega_N(A^{[k]})^2$
1	0	1	0	$(1-\omega_N(A^{[k]}))\omega_N(A^{[k]})^2$
1	1	0	0	$(1-\omega_N(A^{[k]}))\omega_N(A^{[k]})^2$
1	0	0	1	$(1-\omega_N(A^{[k]}))^2\omega_N(A^{[k]})$
0	1	0	1	$(1-\omega_N(A^{[k]}))^2\omega_N(A^{[k]})$
0	0	1	1	$(1-\omega_N(A^{[k]}))^2\omega_N(A^{[k]})$
1	1	1	1	$\omega_N(A^{[k]})^3$

From the four last cases, we have :

$$\omega_N(\bigoplus_{j=0}^{j\leq 2} A_{i+j}^{[k]}) = 3(1-\omega_N(A^{[k]}))^2\omega_N(A^{[k]}) + (\omega_N(A^{[k]}))^3. \tag{22}$$

Which give us, the entropy of the parity :

$$H(\bigoplus_{j=0}^{j\leq 2} A_{i+j}^{[k]}) = H(3(1-\omega_N(A^{[k]}))^2\omega_N(A^{[k]}) + (\omega_N(A^{[k]}))^3). \tag{23}$$

So,

$$H(C_i^{2k+1}) = H(3(1-\omega_N(A^{[k]}))^2\omega_N(A^{[k]}) + (\omega_N(A^{[k]}))^3). \qquad (24)$$

7.3 Comparison between Hs and $H(C_i^{2k+1})$

Obviously, we want th amount of information sacrificed to be greater than the amount of information revealed :

$$H_s \geq H(C_i^{2k+1}). \qquad (25)$$

With (20) and (24), it becomes

$$2H(\omega_N(A^{[k]})) \geq H(3(1-\omega_N(A^{[k]}))^2\omega_N(A^{[k]}) + (\omega_N(A^{[k]}))^3). \qquad (26)$$

The following graph shows $H(C^{2k+1})$ and Hs as functions of p_b.

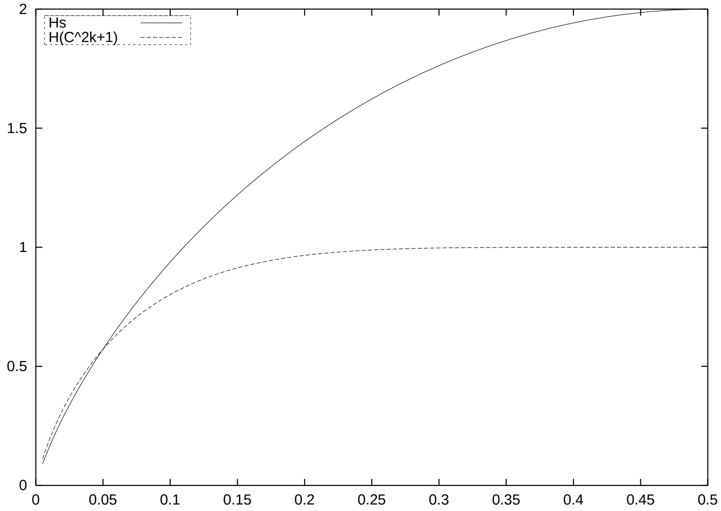

Fig. 2. This graph shows $H(C^{2k+1})$ and Hs as functions of p_b. For $p_b > \frac{1}{20}$, the amount of information revealed is lesser than the amount of information sacrificed

This inequality is true for $\omega_N(A^{[k]}) \in [\frac{1}{20} : \frac{1}{2}]$. To insure the security of the protocol, the inequality must be true for each round of the reconciliation :

$$\forall k \leq r\omega_N(A^{[k]}) \in [\frac{1}{20} : \frac{1}{2}]. \qquad (27)$$

As $\{\omega_N(A^{[k]})\}_{0\leq k\leq r}$ is decreasing and $\omega_N(A^{[0]}) \leq \frac{1}{2}$, we just have to prove that the normalized weight of the residual sequence $A^{[r]}$ after the reconciliation is greater than $\frac{1}{20}$

$$\omega_N(A^{[r]}) \geq \frac{1}{20}. \tag{28}$$

Using (16), this inequality becomes

$$\frac{p_b^2}{(1-p_b)^2+p_b^2} \geq \frac{1}{20}. \tag{29}$$

So,the reconciliation algorithm REC(3,2) is secure if

$$p_b \geq \frac{\sqrt{19}-1}{18}. \tag{30}$$

It means that eve gather no information from the communications C^t between Alice and Bob if the initial normalized weight of the sequences is in the range $[\frac{\sqrt{19}-1}{18} : \frac{1}{2}]$. Under this condition, we have :

$$I(S;C^t) \approx 0. \tag{31}$$

Moreover, as Eve has no initial sequence Z, we can write :

$$I(S;C^tZ) \approx 0. \tag{32}$$

8 Choice of the Parameter p_b

8.1 Constraints on the Choice of p_b

The bias of the random generators used to build $A^{[0]}$ and $B^{[0]}$ is the most important parameter of CHIMERA, as the security and the efficiency of the protocol depend on the value of p_b.

As seen in the proof of the property $|S| \approx H(S)$, the bias p_b should not be greater than $\frac{1}{4}$ to be efficient. Moreover, the proof of the property $I(S;C^tZ) \approx 0$ stands that CHIMERA is safe if p_b is greater than $\frac{\sqrt{19}-1}{18}$. So, the bias of the random generators must be choose in the range $[\frac{\sqrt{19}-1}{18} : \frac{1}{4}]$.

8.2 Simulation Results

We have made simulations with sequences $A^{[0]}$ and $B^{[0]}$ of length $2 \cdot 10^8$ bits. The bias of the random generators is set in the range $[0 : \frac{1}{2})$ (although only the range $[\frac{\sqrt{19}-1}{18} : \frac{1}{4}]$ is really useful in CHIMERA) and the reconciliation round is repeated while Alice and Bob's sequences are different.

Then, we have consider the residual length of the sequences weighted by the entropy of the normalized weight of the sequences, i.e. the length of the sequences compressed with an optimal compression code (like the extended Huffman code). The results of these simulations are presented in the following graph. The x-axis is the bias p_b and the y-axis is the residual length $|S|$.

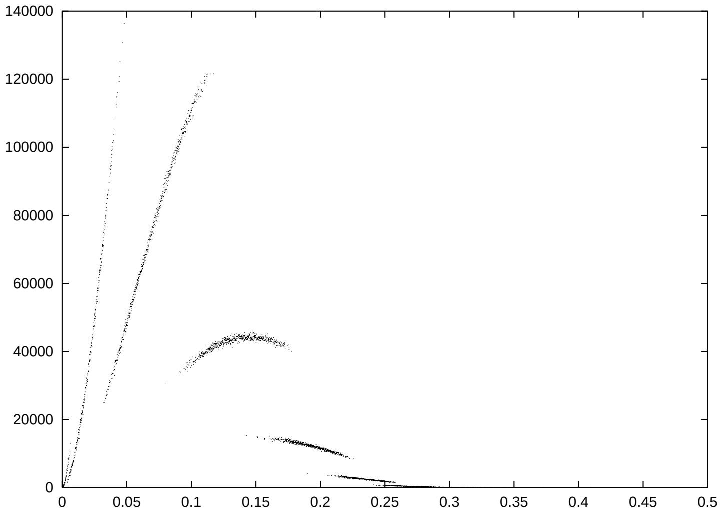

Fig. 3. This graph shows the residual length $|S|$ as a function of p_b. The value p_b we propose is 0.1875

As stands in [3], the goal is to make $|S|$ as large as possible. In the range $[\frac{\sqrt{19}-1}{18} : \frac{1}{4}]$, we have two clouds of points ; the first one, located in $[\frac{\sqrt{19}-1}{18} :\approx 0.22]$, re-groups the results of the simulations with six rounds of reconciliation. The other cloud of points re-groups the results of the simulations with seven rounds of reconciliation.

As one can see, in the range $[\frac{\sqrt{19}-1}{18} :\approx 0.22]$ the residual length $|S|$ is greater than the length $|S|$ in the range $[\approx 0.22 : \frac{1}{4}]$. Moreover, in the first range six rounds of reconciliation, instead of seven rounds, are needed. So we have to chose $p_b \in [\frac{\sqrt{19}-1}{18} :\approx 0.22]$.

Moreover, as the first cloud decreases with p_b, the bias of the random generator should be close to $\frac{\sqrt{19}-1}{18}$. For implementation convenience, we propose to use :

$$p_b = \frac{3}{16}. \tag{33}$$

8.3 Creation of a Biased Random Generator for $p_b = \frac{3}{16}$

The bias p_b can be easily obtain with a combination of non-biased random generators. For example, considering the outputs a, b, c and d of four non-biased random generators, the logical combination

$$p = a \cdot b \cdot c + a \cdot b \cdot d^1. \tag{34}$$

[1] $\cdot$ denotes the logical operator AND, $+$ denotes the logical operator OR

is a biased random generator of bias $p_b = \frac{3}{16}$.

A such simple construction can be implement in any environment and let Alice and Bob build their initial sequences with very light calculus. As the other parts of the protocol need very light calculations (XOR and Huffman coding with pre-calculated trees), our intend is to make the creation of the sequence as easy as the rest of the protocol.

9 Parameter of the Extended Huffman Code

The efficiency of the Huffman code depends on the number of symbols of the language on which is based the Huffman tree. For example if only two symbols appears, whatever their frequencies, the Huffman tree will be a simple root. But, if you consider n-tuples of symbols as the symbols of a language, the Huffman code become more and more efficient as n increases. The compression ratio is, of course, bounded by the entropy of the language.

For the last stage of CHIMERA, we have to find a size of the n-tuples such as a 128 bit key created with CHIMERA as at least 127 bits of entropy. The method to find n is simple : we calculate the minimum-redundancy code for an increasing n, with the algorithm presented in [7] until we found a compression ration $\mathcal{R}_n$ such as :

$$128 \cdot \frac{H(\omega_N(A^{[6]}))}{\mathcal{R}_n} \geq 127. \tag{35}$$

The following table present, for a given n, the compression ratio of the minimal-redundancy code obtained with n-tuples as symbols, and the entropy of a 128 bits key created with this minimal-redundancy code:

Table 5. Compression ratio and entropy of the key for a given length of the extended Huffman code.

n	$\mathcal{R}_n$	H(S)
1	1	36.9
2	0.5745	64.3
3	0.4347	85.1
4	0.3685	100.2
5	0.3378	109.4
6	0.3179	116.3
7	0.3056	121.0
8	0.3007	122.9
9	0.2971	124.4
10	0.2936	125.8
11	0.2905	127.2

The compression ratio for $n = 11$ is close enough to entropy of $H(A^{[6]}) \approx 0.28878$ to obtain a key with an entropy greater than 127.

Considering bigger n-tuples, one has a better approximation of the entropy. Nevertheless, the Huffman tree need more memory. with 11-tuples, the compression table (i.e. the Huffman tree) needs

10 Average Length of the Keys

The length of the keys can be easily calculated knowing the length $|A^{[r]}|$. As we need empirically 6 rounds of reconciliations to have $P(S \neq S') \approx 0$, we set $r = 6$ for $p_b = \frac{3}{16}$.

10.1 Residual Length $|A^{[6]}|$

The amount of bits kept after a reconciliation round is a function of the normalized distance between the sequences : the closer the sequences are, the fewer 3-tuples are destroyed.

As one bit is kept when the 3-tuples have the same parity, and none if the parities differ, noting $R(d_N(A^{[k]}, B^{[k]}) = \frac{A^{[k+1]}}{A^{[k]}}$ the reduction factor of the sequence, we have (with i multiple of 3):

$$R(d_N(A^{[k]}, B^{[k]})) = \frac{P((\bigoplus_{j=0}^{2} A_{i+j}^{[k]}) = (\bigoplus_{j=0}^{2} B_{i+j}^{[k]}))}{3}. \tag{36}$$

Considering the 3-tuples with the same parity, the table in the section (5.3) gives, setting $d_N = d_N(A^{[k]}, B^{[k]})$:

$$R(d_N) = \frac{(1 - d_N)^3 + 3(1 - d_N)d_N^2}{3}. \tag{37}$$

As the reconciliation is an iterative process, the length $|A^{[6]}|$ is reduced six times, with a ratio depending on the normalized distance between Alice and Bob's before each round of reconciliation REC(3,2). so, the length $|A^{[6]}|$ is :

$$|A^{[6]}| = |A^{[0]}| \prod_{i=0}^{5} R(d_N(A^{[k]}, B^{[k]})). \tag{38}$$

Of course, $d_N(A^{[k]}, B^{[k]})$ is given for each iteration by (4).

10.2 Length of the Key S

At the end of the reconciliation, Alice and Bob own respectively the sequences $A^{[6]}$ and $B^{[6]}$, of length $k = |A^{[6]}|$ and of normalized weight $\omega_N(A^{[6]})$. The normalized weight is given by (16) :

$$\omega_N(A^{[6]}) = \frac{(\omega_N(A^{[0]}))^2}{(1 - \omega_N(A^{[0]}))^2 + (\omega_N(A^{[0]}))^2}. \tag{39}$$

These sequences equal with a very high probability are compressed at the end of the protocol with an extended Huffman code which compression ratio is very close to the entropy of the sequence. Thus, the length of the key is :

$$|S| = H(\omega_N(A^{[6]})) \cdot |A^{[6]}|. \tag{40}$$

From (38) and (39), we have :

$$|S| = H(\frac{(\omega_N(A^{[0]}))^2}{(1-\omega_N(A^{[0]}))^2 + (\omega_N(A^{[0]}))^2})|A^{[0]}| \prod_{i=0}^{5} R(d_N(A^{[k]}, B^{[k]})). \tag{41}$$

With the extended Huffman code of length $n = 11$, the practical length of the keys is :

$$|S| = \mathcal{R}_{11}|A^{[0]}| \prod_{i=0}^{5} R(d_N(A^{[k]}, B^{[k]})). \tag{42}$$

The evaluation of this formula gives:

$$|S| \approx 6.37 \cdot 10^{-5}|A^{[0]}|. \tag{43}$$

So Alice and Bob can create a common key of 128 bits with initial sequences of length 2000000 bits.

11 Conclusion

The main points addressed in this paper are :

- A generalized definition of reconciliation has been proposed to let the users destroy more than one symbol of their sequences. The generalization is useful when the entropy of the reconciled sequences is not maximal.
- A unconditionally secure key agreement protocol, called CHIMERA, has been proposed. Its soundness and its security has been proved. The CHIMERA uses no specific devices unlike other unconditionally secure key agreement protocol.
- Convenient parameters has been given for practical implementation of the CHIMERA.

Acknowledgments. We are grateful to Alistair Moffar for the source code of his in-place calculation of minimum redundancy codes which was helpful in the determination of the characteristics of the extended Huffman code. We are also grateful to Sami Harari for his advice in the writing of this paper.

References

1. W. Diffie and M. Hellman. New directions in cryptography, 1976.
2. Charles Bennett, H., François Bessette, Gilles Brassard, and Louis Salvail. Experimental quantum cryptography. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 5(1):3–28, ???? 1992.

3. Ueli Maurer. Information-theoretic cryptography. In Michael Wiener, editor, *Advances in Cryptology - CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 47–64. Springer-Verlag, 1999.
4. Ueli M. Maurer. Information-theoretically secure secret-key agreement by NOT authenticated public discussion. In *Theory and Application of Cryptographic Techniques*, pages 209–225, 1997.
5. Stefan Wolf. Strong security against active attacks in information-theoretic secret-key agreement. In *Advances in Cryptology – ASIACRYPT 98: International Conference on the Theory and Application of Cryptology*, volume 1514 of *Lecture Notes in Computer Science*, pages 405–419. Springer-Verlag, 1998.
6. D. A. Huffman. A method for the construction of minimum redundancy codes. *Proceedings of the Institute of Electronics and Radio Engineers*, 40:1098–1101, 1952.
7. A. Moffat and J. Katajainen. In-place calculation of minimum-redundancy codes. In S.G. Akl, F. Dehne, and J.-R. Sack, editors, *Proc. Workshop on Algorithms and Data Structures*, pages 393–402, Queen's University, Kingston, Ontario, August 1995. LNCS 955, Springer-Verlag.