Chapter 4

Collusion 3-Secure Fingerprinting Codes

In this chapter we present a construction to come up with collusion-secure fingerprinting codes for collusions of up to 3 colluders. Results presented here have been published in [SD02b]. For a not too large number of buyers, our construction generates much shorter codes than those obtained from the general construction [BS95] for $c = 3$. The basic idea is to compose a new kind of code, which we call *scattering code*, with a *dual binary Hamming code*.

Section 4.1 presents some definitions and properties on dual Hamming codes. Section 4.2 presents a set of lemmas on the probability of successful collusion as a function of colluders'strategy. The construction and decoding of scattering codes are introduced in Section 4.3. Then, Section 4.4 explains how to generate fingerprinting codes secure against collusions of up to three buyers by composing a scattering code code with a dual binary Hamming code together with some numerical results comparing the length of codes from our construction with the length of codes from [BS95].

4.1 Dual binary Hamming codes

The dual code of a binary Hamming code (denoted by $DH(n)$) is a binary code with 2^n codewords of length $N = 2^n - 1$ such that the distance between any two codewords is 2^{n-1}. A few definitions and useful properties related to such codes are presented next.

Definition 1 *Let a^1, a^2, a^3 be three codewords of a $DH(n)$ code,* i.e. *$a^i = a^i_1 a^i_2 \cdots a^i_N$. Define $inv(a^1, a^2, a^3)$ to be the set of invariant positions between all three codewords, that is, those bit positions in which all three codewords have the same bit value. Formally speaking,*

$$inv(a^1, a^2, a^3) = \{i,\ 1 \leq i \leq N,\ a^1_i = a^2_i = a^3_i\}$$

Definition 2 *Let a^1, a^2, a^3 be three codewords of a $DH(n)$ code. Define $minor(a^1; a^2, a^3)$ to be the set of bit positions in which a^1 has a value different from the values in a^2 and a^3 (for such positions, $a^2_i = a^3_i$). Formally speaking,*

$$minor(a^1; a^2, a^3) = \{i,\ 1 \leq i \leq N,\ a^1_i \neq a^2_i,\ a^1_i \neq a^3_i\}$$

Lemma 1 *Let a^1, a^2, a^3 be three codewords of a $DH(n)$ code and let $|\cdot|$ denote the bitlength operator. Then it holds that $|inv(a^1, a^2, a^3)| = 2^{n-2} - 1$, $|minor(a^1; a^2, a^3)| = 2^{n-2}$, $|minor(a^2; a^1, a^3)| = 2^{n-2}$ and $|minor(a^3; a^1, a^2)| = 2^{n-2}$.*

Proof: Let a^1, a^2, a^3 be three codewords of a $DH(n)$ code. Define $I = inv(a^1, a^2)$ and $\overline{I}$ to be the positions not in I. Since $d(a^i, a^j)_{i \neq j} = 2^{n-1}$, then $|I| = 2^{n-1} - 1$.

Let $x = |inv(a^1, a^2, a^3)|$ (obviously, $inv(a^1, a^2, a^3) \subset I$) and let y be the total number of positions $i \in \overline{I}$ where $a^2_i = a^3_i$ (these are the positions that

form $minor(a^1; a^2, a^3)$). As $d(a^2, a^3) = 2^{n-1}$, then $x + y = 2^{n-1} - 1$.

There are $2^{n-1} - 1 - x$ positions $i \in I$ where $a_i^3 \neq a_i^1$ (these are the positions that form $minor(a^3; a^1, a^2)$) and y positions $i \in \overline{I}$ where $a_i^3 \neq a_i^1$. As $d(a^3, a^1) = 2^{n-1}$ then $2^{n-1} - 1 - x + y = 2^{n-1}$.

Solving the following equations for x and y

$$\begin{cases} x + y = 2^{n-1} - 1 \\ 2^{n-1} - 1 - x + y = 2^{n-1} \end{cases}$$

we get $x = 2^{n-2} - 1$ and $y = 2^{n-2}$. Finally, we conclude

$|inv(a^1, a^2, a^3)| = x = 2^{n-2} - 1$,

$|minor(a^1; a^2, a^3)| = y = 2^{n-2}$,

$|minor(a^2; a^1, a^3)| = 2^{n-1} - y = 2^{n-2}$,

$|minor(a^3; a^1, a^2)| = 2^{n-1} - 1 - x = 2^{n-2}$

□

Example: The following are three codewords of a $DH(5)$ code.

	$inv(a^1, a^2, a^3)$	$minor(a^1; a^2, a^3)$	$minor(a^2; a^1, a^3)$	$minor(a^3; a^1, a^2)$
a^1	0000000	11111111	00000000	11111111
a^2	0000000	00000000	11111111	11111111
a^3	0000000	00000000	00000000	00000000

The codeword length is $2^5 - 1 = 31$. Now $|inv(a^1, a^2, a^3)| = 2^{5-2} - 1 = 7$, $|minor(a^1; a^2, a^3)| = |minor(a^2; a^1, a^3)| = |minor(a^3; a^1, a^2)| = 2^{5-2} = 8$.

□

Lemma 2 *Let a^1, a^2, a^3 be three codewords of a $DH(n)$ code. Then it holds that:*

- *There exists one and only one codeword $a^z \in DH(n) \backslash \{a^1, a^2, a^3\}$*

such that $a_i^z = a_i^1 = a_i^2 = a_i^3$, $\forall i \in inv(a^1, a^2, a^3)$. *Furthermore,* $a_i^z = a_i^1$, $\forall i \in minor(a^1; a^2, a^3)$, $a_i^z = a_i^2$, $\forall i \in minor(a^2; a^1, a^3)$ *and* $a_i^z = a_i^3$, $\forall i \in minor(a^3; a^1, a^2)$.

- *The remaining codewords satisfy that* $\forall a^j \in DH(n) \backslash \{a^1, a^2, a^3, a^z\}$, $d_{inv(a^1,a^2,a^3)}(a^j, a^1) = d_{minor(a^1;a^2,a^3)}(a^j, a^1) = d_{minor(a^2;a^1,a^3)}(a^j, a^1) = d_{minor(a^3,a^1,a^2)}(a^j, a^1) = 2^{n-3}$, *where* $d_P(x, y)$ *denotes Hamming distance between codewords* x *and* y *restricted to bit positions in* P. *The same distances hold with respect to* a^2 *and* a^3.

Proof: First, the existence and properties of a^z will be proven. As a $DH(n)$ code is a linear code, any linear combination of codewords results in another codeword. Then, we get $a^z = a^1 \oplus a^2 \oplus a^3$, where $\oplus$ denotes the component-wise modulo 2 addition.

We prove that $a_i^z = a_i^1 = a_i^2 = a_i^3$, $\forall i \in inv(a^1, a^2, a^3)$. This is true because if $a_i^1 = a_i^2 = a_i^3 = 1$, then $a_i^1 \oplus a_i^2 \oplus a_i^3 = 1$, and if $a_i^1 = a_i^2 = a_i^3 = 0$, then $a_i^1 \oplus a_i^2 \oplus a_i^3 = 0$.

Then, we prove $a_i^z = a_i^1$, $\forall i \in minor(a^1; a^2, a^3)$. This is true because $a_i^z = a_i^1 \oplus a_i^2 \oplus a_i^3$ and as $a_i^2 = a_i^3$, then $a_i^z = a_i^1$.

Using the same idea, we can prove $a_i^z = a_i^2$, $\forall i \in minor(a^2; a^1, a^3)$ and $a_i^z = a_i^3$, $\forall i \in minor(a^3; a^1, a^2)$.

Next, the second part of the Lemma will be proven. Consider $a^j \in DH(n) \backslash \{a^1, a^2, a^3, a^z\}$

Call x the number of positions in $inv(a^1, a^2, a^3)$ where $a_i^j = a_i^1$. Then the number of positions in $inv(a^1, a^2, a^3)$ where $a_i^j \neq a_i^1$ is $2^{n-2} - 1 - x$ (see Lemma 1).

Call y the number of positions in $minor(a^1; a^2, a^3)$ where $a_i^j = a_i^1$. Then the number of positions in $minor(a^1; a^2, a^3)$ where $a_i^j \neq a_i^1$ is $2^{n-2} - y$.

Call z the number of positions in $minor(a^2; a^1, a^3)$ where $a_i^j = a_i^1$. Then the number of positions in $minor(a^2; a^1, a^3)$ where $a_i^j \neq a_i^1$ is $2^{n-2} - z$.

Call t the number of positions in $minor(a^3; a^1, a^2)$ where $a_i^j = a_i^1$. Then the number of positions in $minor(a^3; a^1, a^2)$ where $a_i^j \neq a_i^1$ is $2^{n-2} - t$.

Since $d(a^j, a^1) = d_{inv(a^1,a^2,a^3)}(a^j, a^1) + d_{minor(a^1;a^2,a^3)}(a^j, a^1) + d_{minor(a^2;a^1,a^3)}(a^j, a^1) + d_{minor(a^3,a^1,a^2)}(a^j, a^1) = 2^{n-1}$, we have

$$(2^{n-2} - 1 - x) + (2^{n-2} - y) + (2^{n-2} - z) + (2^{n-2} - t) = 2^{n-1}$$

Since $d(a^j, a^2) = d_{inv(a^1,a^2,a^3)}(a^j, a^2) + d_{minor(a^1;a^2,a^3)}(a^j, a^2) + d_{minor(a^2;a^1,a^3)}(a^j, a^2) + d_{minor(a^3,a^1,a^2)}(a^j, a^2) = 2^{n-1}$, we have

$$(2^{n-2} - 1 - x) + y + z + (2^{n-2} - t) = 2^{n-1}$$

Since $d(a^j, a^3) = d_{inv(a^1,a^2,a^3)}(a^j, a^3) + d_{minor(a^1;a^2,a^3)}(a^j, a^3) + d_{minor(a^2;a^1,a^3)}(a^j, a^3) + d_{minor(a^3,a^1,a^2)}(a^j, a^3) = 2^{n-1}$, we have

$$(2^{n-2} - 1 - x) + y + (2^{n-2} - z) + t = 2^{n-1}$$

Since $d(a^j, a^z) = d_{inv(a^1,a^2,a^3)}(a^j, a^z) + d_{minor(a^1;a^2,a^3)}(a^j, a^z) + d_{minor(a^2;a^1,a^3)}(a^j, a^z) + d_{minor(a^3,a^1,a^2)}(a^j, a^z) = 2^{n-1}$, we have

$$(2^{n-2} - 1 - x) + (2^{n-2} - y) + z + t = 2^{n-1}$$

From the expressions above, the following equation system can be derived:

$$\begin{cases} x+y+z+t = 2^{n-1}-1 \\ -x+y+z-t = 1 \\ -x+y-z+t = 1 \\ -x-y+z+t = 1 \end{cases}$$

By solving it, we get $x = 2^{n-3}-1$ and $y = z = t = 2^{n-3}$.

Finally, we conclude,

$d_{inv(a^1,a^2,a^3)}(a^j, a^1) = 2^{n-2} - 1 - x = 2^{n-3}$

$d_{minor(a^1;a^2,a^3)}(a^j, a^1) = 2^{n-2} - y = 2^{n-3}$

$d_{minor(a^2;a^1,a^3)}(a^j, a^1) = 2^{n-2} - z = 2^{n-3}$

$d_{minor(a^3;a^1,a^2)}(a^j, a^1) = 2^{n-2} - t = 2^{n-3}$

In the same way, we can prove these distances hold between a^j and a^2, a^3.

□

Example: The table below displays the unique codeword a^z corresponding to three particular codewords a^1, a^2, a^3 of a $DH(5)$ code.

	$inv(a^1,a^2,a^3)$	$minor(a^1;a^2,a^3)$	$minor(a^2;a^1,a^3)$	$minor(a^3;a^1,a^2)$
a^1	0000000	11111111	00000000	11111111
a^2	0000000	00000000	11111111	11111111
a^3	0000000	00000000	00000000	00000000
a^z	0000000	11111111	11111111	00000000

It can be seen that $a_i^z = a_i^1 = a_i^2 = a_i^3,\ \forall i \in inv(a^1,a^2,a^3)$. Also, $a_i^z = a_i^1,\ \forall i \in minor(a^1;a^2,a^3)$, $a_i^z = a_i^2,\ \forall i \in minor(a^2;a^1,a^3)$ and $a_i^z = a_i^3,\ \forall i \in minor(a^3;a^1,a^2)$.

□

Example: The table below displays three codewords a^1, a^2, a^3 of a $DH(5)$ code and another codeword $a^i \in DH(5)\backslash\{a^1, a^2, a^3, a^z\}$.

	$inv(a^1, a^2, a^3)$	$minor(a^1; a^2, a^3)$	$minor(a^2; a^1, a^3)$	$minor(a^3; a^1, a^2)$
a^1	0000000	11111111	00000000	11111111
a^2	0000000	00000000	11111111	11111111
a^3	0000000	00000000	00000000	00000000
a^i	0001111	00001111	00001111	00001111

It can be seen that $d_{inv(a^1,a^2,a^3)}(a^i, a^1) = d_{minor(a^1;a^2,a^3)}(a^i, a^1) = d_{minor(a^2;a^1,a^3)}(a^i, a^1) = d_{minor(a^3,a^1,a^2)}(a^i, a^1) = 2^{n-3} = 4$. The same distances hold between a^i and a^2, a^3. □

4.2 3-Collusions over $DH(n)$

4.2.1 Detectable positions

Let us assume three dishonest buyers c^1, c^2, c^3 compare their copies of the same multimedia content. According to the marking assumption [BS95], they can only modify the embedded marks in those *detectable* positions where not all three marks take the same bit value. In those positions, colluders can set the corresponding bit to '0', '1' or "unreadable". In this way, we conclude that, if three different buyers are assigned codewords a^1, a^2 and a^3 of a $DH(n)$ code, the result of their collusion will be a word a^{coll} where no bit has been modified in the $2^{n-2} - 1$ positions in $inv(a^1, a^2, a^3)$. On the other hand, colluders will be able to detect and identify positions in $minor(a^1; a^2, a^3)$ as the bit positions of those content fragments which are identical between the copies of c^2 and c^3 and different from the copy of c^1. In a similar way,

$minor(a^2; a^1, a^3)$ and $minor(a^3; a^1, a^2)$ can be detected and identified as well.

4.2.2 Decoding by minimum distance

As it has been said, colluders can generate a new object whose embedded codeword may have been altered in detectable positions. In this way, it is possible that the word retrieved from a collusion-generated object does not correspond to any $DH(n)$ codeword. In such cases, the recovered word will be error-corrected by minimum distance.

Thus, in order for a collusion to be successful, colluders c^1, c^2, c^3 with assigned codewords a^1, a^2, a^3, respectively, must generate, by mixing fragments of their copies, a word such that the closest codeword in the $DH(n)$ code is not in $\{a^1, a^2, a^3\}$ (see Figure 4.1). A successful collusion will cause an innocent buyer to be accused in lieu of the colluders. Note that we are assuming that colluders do not generate "unreadable" positions when colluding over $DH(n)$ codewords. It will be shown later that our construction actually prevents unreadable positions from being fed by colluders to the dual Hamming decoder.

4.2.3 The aim of colluders

As decoding is done by minimum distance, the aim of colluders is to come up with an object whose embedded word is as distant as possible from their assigned codewords.

Intuitively, it can be realized that all colluders must contribute with the same number of bits from their corresponding codewords. Otherwise, the collusion-generated word would be closer to the codewords of those colluders having contributed more bits.

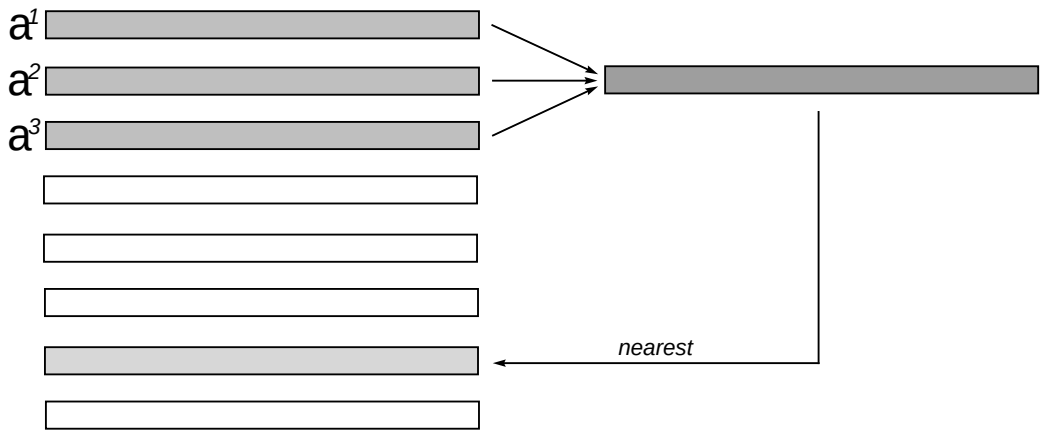

Figure 4.1: A successful collusion.

Definition 3 *A* p-majority *collusion strategy is one in which colluders choose with probability* p *the majority bit value in positions* $minor(a^i; a^j, a^k)$ *(that is, the bit values in* a^j *or* a^k*) (See Figure 4.2).*

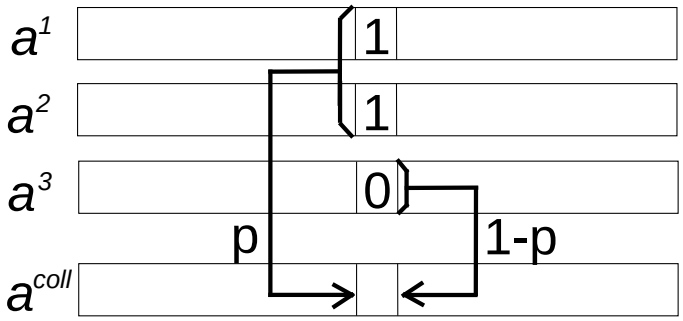

Figure 4.2: p-majority collusion strategy.

It can be seen that a word generated using a p-majority strategy from $a^1, a^2, a^3 \in DH(n)$ is expected to have the same distance to a^1, a^2 and a^3.

4.2.4 Distance from a collusion-generated word to colluders'codewords

Lemma 3 *Let a^{coll} be a word that has been generated using a p-majority collusion strategy between three codewords $a^1, a^2, a^3 \in DH(n)$. It holds that $d_1 = d(a^{coll}, a^i) = K_1, \forall i = 1, 2, 3$ with*

$$p_1(k) = p(K_1 = k) = \sum_{t=\max(0,k-2^{n-1})}^{\min(k,2^{n-2})} b(t; 2^{n-2}, p) b(k-t; 2^{n-1}, 1-p)$$

where $b(x_1; x_2, x_3)$ is the binomial probability function (x_2 is the number of trials, x_3 the success probability per trial and x_1 es the number of successful trials).

Proof: Without loss of generality, take $i = 1$. We have that, for bit positions in $inv(a^1, a^2, a^3)$, there is no difference between a^1 and a^{coll} since bits in those positions are undetectable. Also, each of the 2^{n-2} bits in $minor(a^1; a^2, a^3)$ differs between a^1 and a^{coll} with probability p; therefore, the probability of there being t differing bits in those positions is given by a binomial probability function $b(t; 2^{n-2}, p)$. Also, each of the $2 \cdot 2^{n-2}$ bits in $minor(a^2; a^1, a^3)$ and $minor(a^3; a^1, a^2)$ differs between a^1 and a^{coll} with probability $(1-p)$; therefore, the probability of there being $k-t$ differing bits in those positions is given by a binomial probability function $b(k-t; 2^{n-1}, 1-p)$. In this way, the expression in the lemma corresponds to the probability of there being a total of $t + (k-t) = k$ differing bits between a^1 and a^{coll}. □

Remarks: The total amount of differing bits is the addition of two binomially distributed random variables. We use this fact to compute its

expected value as

$$E(d_1) = p \cdot 2^{n-2} + (1-p)2^{n-1} = 2^{n-1} - p \cdot 2^{n-2}$$

As can be seen in Table 4.1, the expected number of differing bits between the word (a^{coll}) generated by collusion and any of the colluders' codewords ($a^1, a^2, a^3 \in DH(6)$) decreases as the value p gets closer to 1 (a^{coll} gets closer to a^1, a^2, a^3).

p	0	0.2	0.4	0.6	0.8	1
$E(d_1)$	32	28.8	25.6	22.4	19.2	16

Table 4.1: Expected number of differing bits between a word a^{coll} generated using a p-majority strategy and any of the colluders'codewords. The code is a $DH(6)$.

Lemma 4 *Let a^{coll} be a word generated using a p-majority collusion strategy between three codewords $a^1, a^2, a^3 \in DH(n)$. It holds that $d_2 = \min_{i=1,2,3} d(a^{coll}, a^i) = K_2$ with*

$$p_2 = p(K_2 = k) = \sum_{i=1}^{3} \binom{3}{i} p_1(k)^i \Big[\sum_{k'>k} p_1(k') \Big]^{3-i}$$

Proof: The expression in the lemma corresponds to the probability of one, two or three codewords in $\{a^1, a^2, a^3\}$ being at distance k from a^{coll} and the remaining codewords being at a greater distance. □

Lemma 5 *Let a^{coll} be a word generated using a p-majority collusion strategy between three codewords $a^1, a^2, a^3 \in DH(n)$. It holds that $d_3 =$*

$\max_{i=1,2,3} d(a^{coll}, a^i) = K_3$ *with*

$$p_3 = p(K_3 = k) = \sum_{i=1}^{3} \begin{pmatrix} 3 \\ i \end{pmatrix} p_1(k)^i \Big[\sum_{k'<k} p_1(k') \Big]^{3-i}$$

Proof: The expression in the lemma corresponds to the probability of one, two or three codewords in $\{a^1, a^2, a^3\}$ being at distance k from a^{coll} and the remaining codewords being at a minor distance. □

Table 4.2 presents expected values of d_2 and d_3 for different values of p over a $DH(6)$ code.

p	0	0.2	0.4	0.6	0.8	1
$E(d_2)$	32	26.5	22.7	19.5	16.9	16
$E(d_3)$	32	31.12	28.46	25.27	21.54	16

Table 4.2: Expected number of differing bits between a word a^{coll} generated using a p-majority strategy and the nearest ($E(d_2)$) and the farthest ($E(d_3)$) among the colluders' codewords. The code is a $DH(6)$.

4.2.5 Distance from a collusion-generated word to codewords not in the collusion

Lemma 6 *Let a^{coll} be a word generated using a p-majority strategy between three codewords $a^1, a^2, a^3 \in DH(n)$ and let a^z be the only codeword in $DH(n) \backslash \{a^1, a^2, a^3\}$ with $a^z_i = a^1_i = a^2_i = a^3_i$, $\forall i \in inv(a^1, a^2, a^3)$ (existence and uniqueness of a^z are guaranteed by Lemma 2). Then, $d_4 = d(a^z, a^{coll}) = K_4$ with*

$$p_4(k) = p(K_4 = k) = b(k; 3 \cdot 2^{n-2}, p)$$

Proof: Lemma 2 says that bits of a^z are identical to bits of a^i in the positions in $minor(a^i; a^j, a^k)$ for $(i,j,k) \in \{(1,2,3),(2,1,3),(3,1,2)\}$. Therefore, the probability of there being k different bits in those $3 \cdot 2^{n-2}$ positions is given by a binomial probability function $b(k; 3 \cdot 2^{n-2}, p)$. □

Remarks: The expected number of differing bits between a^z and a^{coll} is

$$E(d_4) = p \cdot 3 \cdot 2^{n-2}$$

Lemma 7 *Let a^{coll} be a word generated using a p-majority strategy between three codewords $a^1, a^2, a^3 \in DH(n)$ and let a^z be the only codeword in $DH(n)\backslash\{a^1, a^2, a^3\}$ with $a_i^z = a_i^1 = a_i^2 = a_i^3$, $\forall i \in inv(a^1, a^2, a^3)$. Then, for any codeword $a \in DH(n)\backslash\{a^1, a^2, a^3, a^z\}$ it holds that $d_5 = d(a, a^{coll}) = 2^{n-3} + K_5$ with*

$$p_5(k) = p(K_5 = k) = \sum_{t=\max(0,k-3\cdot 2^{n-3})}^{\min\{k,3\cdot 2^{n-3}\}} b(t; 3 \cdot 2^{n-3}, 1-p) b(k-t; 3 \cdot 2^{n-3}, p)$$

Proof: According to Lemma 2, a^{coll} and a have 2^{n-3} differing bits in positions in $inv(a^1, a^2, a^3)$. In each $minor(a^i; a^j, a^k)$, for $(i,j,k) \in \{(1,2,3),(2,1,3),(3,1,2)\}$, a^{coll} has all 2^{n-3} bits each of which is different with probability p and 2^{n-3} bits each of which is different with probability $(1-p)$. Therefore, we have $3 \cdot 2^{n-3}$ bits with probability p of being different, and thus the probability that t of such bits are different is $b(t; 3 \cdot 2^{n-3}, p)$. On the other hand, we have $3 \cdot 2^{n-3}$ bits with probability $1-p$ of being different, and thus the probability that $k-t$ of such bits are different is $b(k-t; 3 \cdot 2^{n-3}, 1-p)$. In this way, the expression in the lemma computes the probability of there being $t + (k-t) = k$ differing bits between a and

a^{coll}. □

Remarks: The expected number of differing bits between a and a^{coll} is

$$E(d_5) = 2^{n-3} + 3 \cdot 2^{n-3}(1-p) + 3 \cdot 2^{n-3}p = 2^{n-1}$$

Table 4.3 presents expected values of d_4 and d_5 for different values of p over a $DH(6)$ code.

p	0	0.2	0.4	0.6	$0.\hat{6}$	0.8	1
$E(d_4)$	0	9.6	19.2	28.8	32	38.4	48
$E(d_5)$	32	32	32	32	32	32	32

Table 4.3: Expected number of differing bits between a word a^{coll} generated using a p-majority strategy and a^z $(E(d_4))$ and the remaining codewords in $DH(6)\backslash\{a^1, a^2, a^3, a^z\}$ $(E(d_5))$.

For the sake of simplicity, let us assume in what follows that d_4 is distributed like d_5. Since for $p > 0.\hat{6}$ the number of differing bits expected for d_4 is greater than the number of different bits expected for d_5 ($E(d_4) > E(d_5) \Leftrightarrow p \cdot 3 \cdot 2^{n-2} > 2^{n-1} \Leftrightarrow p > 0.\hat{6}$), such a distributional assumption will cause actual security to be even slightly higher than computed in what follows.

Lemma 8 *Let a^{coll} be a word generated using a p-majority strategy ($p > 0.\hat{6}$) between three codewords $a^1, a^2, a^3 \in DH(n)$. It holds that $d_6 = \min_{i \notin \{1,2,3\}}\{d(a^{coll}, a^i)\} = 2^{n-3} + K_6$, with*

$$p_6(k) = p(K_6 = k) = \sum_{i=1}^{2^n-3} \binom{2^n - 3}{i} p_5(k)^i \Big[\sum_{k'>k} p_5(k')\Big]^{2^n-3-i}$$

Proof: The expression in the lemma computes the probability that at least one out of the $2^n - 3$ codewords in $DH(n)\backslash\{a^1, a^2, a^3\}$ is at distance k of a^{coll}, with the remaining codewords at a longer distance. □

Table 4.4 presents expected values of d_6 for different values of p over a $DH(6)$ code.

p	$0.\hat{6}$	0.8	1
$E(d_6)$	24.5	25.6	32

Table 4.4: Expected number of differing bits between a word a^{coll} generated using a p-majority strategy ($p > 0.\hat{6}$) and the nearest of the codewords not in the collusion. The code is a $DH(6)$.

As it can be seen in Figure 4.3, when $p > 0.\hat{6}$, d_2 tends to take smaller values than d_6. This means that, with high probability, the codeword in $DH(n)$ nearest to the collusion generated word is a colluder codeword.

4.2.6 Identifying colluders'codewords

Lemma 9 *Let a^{coll} be a word generated using a p-majority strategy ($p > 0.\hat{6}$) between three codewords $a^1, a^2, a^3 \in DH(n)$. The probability that the codeword in $DH(n)$ closest to a^{coll} is* not *in $\{a^1, a^2, a^3\}$ is expressed by*

$$\epsilon = \sum_{k=0}^{2^n-1} p(d_2 = k)p(d_6 \leq k)$$

ϵ is the probability that decoding a^{coll} yields as a result a codeword different from any of the colluders' codewords, that is, the probability of a honest buyer being unjustly accused instead of the colluders.

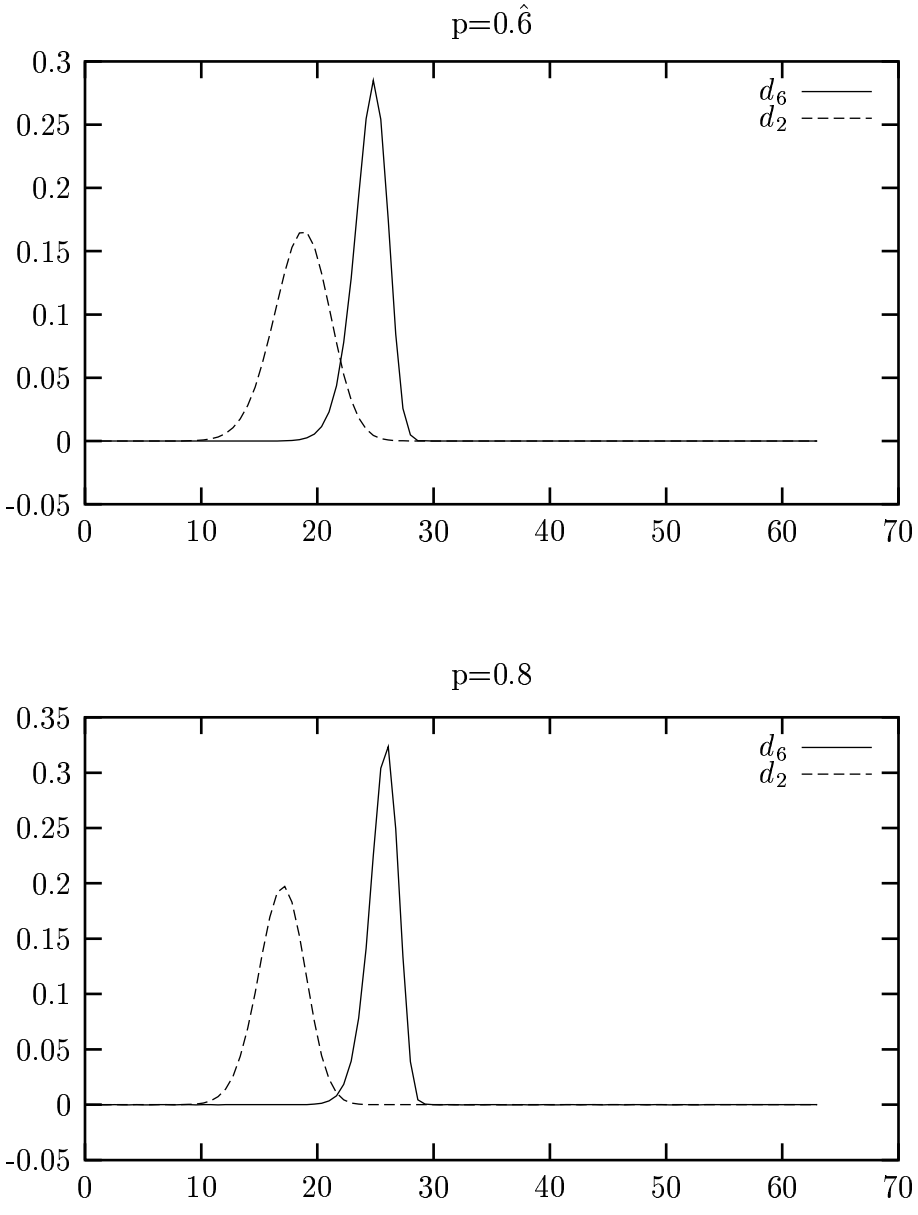

Figure 4.3: Distribution of d_2 and d_6 for $p = 0.\hat{6}$ and $p = 0.8$. The code is a $DH(6)$.

	p					
	0.0	$0.\hat{6}$	0.7	0.8	0.9	1.0
$DH(7)$	1.0	$0.59 \cdot 10^{-3}$	$0.14 \cdot 10^{-3}$	$0.14 \cdot 10^{-6}$	$0.77 \cdot 10^{-14}$	0.0
$DH(8)$	1.0	$0.17 \cdot 10^{-7}$	$0.10 \cdot 10^{-7}$	$0.15 \cdot 10^{-13}$	$0.70 \cdot 10^{-28}$	0.0

Table 4.5: Probability ϵ of success of a 3-collusion in $DH(7)$ and $DH(8)$ for several values of p

Remarks: It can be observed from Table 4.5 that, as n increases and p approaches 1, the probability ϵ of accusing an innocent buyer can be made arbitrarily close to 0.

Lemma 10 *Let a^{coll} be a word generated using a p-majority strategy ($p > 0.\hat{6}$) between three codewords $a^1, a^2, a^3 \in DH(n)$. The probability that the three closest codewords in $DH(n)$ to a^{coll} are $\{a^1, a^2, a^3\}$ is expressed by*

$$1 - \epsilon_2 = \sum_{k=0}^{2^n-1} p(d_3 = k)p(d_6 > k)$$

	p					
	0.0	$0.\hat{6}$	0.7	0.8	0.9	1.0
$DH(7)$	1.0	0.1	$0.5 \cdot 10^{-1}$	$0.1 \cdot 10^{-2}$	$0.25 \cdot 10^{-7}$	0.0
$DH(8)$	1.0	$0.14 \cdot 10^{-2}$	$0.26 \cdot 10^{-3}$	$0.6 \cdot 10^{-7}$	$0.2 \cdot 10^{-16}$	0.0

Table 4.6: Probability ϵ_2 of *not* identifying all three colluders in $DH(7)$ and $DH(8)$ for several values of p

Remarks: It can be observed from table 4.6 that as n increases and p approaches 1, the probability of not identifying all three colluders can be made arbitrarily close to 0.

The problem is that the parameter p defining the collusion strategy is chosen by the colluders, which implies they can take $p = 0$ to make sure they

are not identified!

In Section 4.3, a new kind of codes named *scattering codes* are presented. These codes are used in Section 4.4 to prevent colluders from avoiding identification in this way.

4.3 Scattering codes

In this section, we present a new kind of codes named *scattering codes*. Their construction, decoding and properties have been published in [SD02a].

4.3.1 Construction

A *scattering code* $SC(d,t)$ with parameters (d,t) is defined as a binary code consisting of $2t$ codewords of length $(2t+1)d$ constructed as follows:

1. The construction starts with generation of $SC(1,t)$:
   (a) The i-th codeword for $1 \leq i \leq t$ is constructed by setting the first and the $(i+1)$-th bits of the codeword to '1'. The remaining bits are set to '0'.
   (b) The i-th codeword for $t+1 \leq i \leq 2t$ is constructed by setting the $(i+1)$-th bit of the codeword to '1'. The remaining bits are set to '0'.
2. The code $SC(d,t)$ is generated by replicating d times every bit of $SC(1,t)$. Define a *block* to be a group of d replicated bits.
3. By convention, the first t codewords of $SC(d,t)$ are defined to encode a '1' and the last t codewords are defined to encode a '0'. The first block of the code is called 'Zone-A', the next t blocks are called 'Zone-B' and the last t blocks are called 'Zone-C'.

Example: The following are the codewords of a scattering code $SC(4,3)$.

Encodes	Zone-A	Zone-B			Zone-C		
'1'	1111	1111	0000	0000	0000	0000	0000
	1111	0000	1111	0000	0000	0000	0000
	1111	0000	0000	1111	0000	0000	0000
'0'	0000	0000	0000	0000	1111	0000	0000
	0000	0000	0000	0000	0000	1111	0000
	0000	0000	0000	0000	0000	0000	1111

Using a scattering code, a '1' is encoded by randomly choosing one of the first t codewords and a '0' is encoded by randomly choosing one of the last t codewords.

4.3.2 Decoding

In a scattering code, a word is decoded by using the first applicable rule among the following ordered list:

1. If all bits in 'Zone-A' are '1' and all bits in 'Zone-C' are '0', decode as '1'.
2. If all bits in 'Zone-A' are '0' and all bits in 'Zone-B' are '0', decode as '0'.
3. If in two blocks of 'Zone-B' there is at least one bit in each with value '1', decode as '1'.
4. If in two blocks of 'Zone-C' there is at least one bit in each with value '1', decode as '0'.
5. If there are more '1' bits than '0' bits in 'Zone-A', decode as '1'.

6. If there are more '0' bits than '1' bits in 'Zone-A', decode as '0'.

7. Decode as 'Unreadable'

Note: It is easy to see that an odd value for d makes Rule 7 unreachable, making a '0' or '1' to be always returned.

4.3.3 Collusions over $SC(d,t)$

Lemma 11 *Let b^{coll} be a word generated using a p-majority strategy between three codewords $b^1, b^2, b^3 \in SC(d,t)$ encoding the same bit value v. Then, b^{coll} decodes as v with probability 1.*

Proof: It can be seen that, if v ='1', bits in 'Zone-A' and in 'Zone-C' stay undetectable and thus decoding will be through Rule 1 and return a value '1'.

If v ='0', bits in 'Zone-A' and in 'Zone-B' also stay undetectable. Thus, decoding will be through Rule 2 and return a value '0'. □

Lemma 12 *Let b^{coll} be a word generated using a p-majority strategy between three codewords $b^1, b^2, b^3 \in SC(d,t)$, with two of them ($b^1$ and b^2) encoding a value v and the other (b^3) the value $\overline{v}$. Then, the probability that b^{coll} decodes as v is given by*

$$p(v) = (1 - \frac{1}{t})p_{dif}(v) + \frac{1}{t}p_{coi}(v)$$

where $p_{dif}(v)$ is the probability of decoding as v when $b^1 \neq b^2$ and is computed as $p_{dif}(v) = 1 - p_{dif}(\overline{v})$ (we assume d to have an odd value) and

$$\begin{aligned} p_{dif}(\overline{v}) &= (1-p)^d p^{2d} + \\ &+ 2 \cdot p^d (1-p^d) \sum_{k=0}^{\lfloor \frac{d-1}{2} \rfloor} b(k;d,p) + \\ &+ p^{2d} \sum_{k=1}^{\lfloor \frac{d-1}{2} \rfloor} b(k;d,p) \end{aligned}$$

and $p_{coi}(v)$ *is the probability of decoding as* v *when* $b^1 = b^2$ *and is computed as*

$$\begin{aligned} p_{coi}(v) &= p^{2d} + \\ &+ (1-p^d)\sum_{k=\lfloor\frac{d+2}{2}\rfloor}^{d} b(k;d,p) + \\ &+ p^d \sum_{k=\lfloor\frac{d+2}{2}\rfloor}^{d-1} b(k;d,p) \end{aligned}$$

Proof: In a collusion between three codewords $b^1, b^2, b^3 \in SC(d,t)$ with two of them (b^1 and b^2) encoding a value v (without loss of generality, assume $v = 1$ and $\overline{v} = 0$), we have $b^1 = b^2$ with probability $\frac{1}{t}$ and $b^1 \neq b^2$ with probability $1 - \frac{1}{t}$.

a) In the case $b^1 \neq b^2$, we compute $p_{dif}(v) = 1 - p_{dif}(\overline{v})$ (we assume d to have an odd value), where $p_{dif}(\overline{v})$ corresponds to the probability of decoding $\overline{v}$ after a collusion based on a p-majority strategy. $p_{dif}(\overline{v})$ is actually the probability of decoding using Rules 2 or 6.

   - Rule 2 will be applied if all bits in 'Zone-A' and 'Zone-B' are '0'. Since we are assuming a p-majority strategy, all bits in 'Zone-A' will be '0' with probability $b(d;d,1-p) = (1-p)^d$, because the majority bit in these positions is '1'. Since $b1 \neq b2$, there will be two detectable blocks in 'Zone-B' where the majority bit is '0'. Bits in 'Zone-B' will be all '0' with probability $b(2d;2d,p) = p^{2d}$. So, the probability of applying Rule 2 is $(1-p)^d p^{2d}$.
   - Since only one out of the three colluding codewords has value '0', it is not possible to have more than one block of 'Zone-C' with bit values different from '0'. So Rule 4 cannot be applied.
   - The next possibility for decoding as '0' is to apply Rule 6. This happens if there are more '0' bits than '1' bits in 'Zone-A' and no other rule between 1 and 5 has been applied before. In order to

render Rule 3 not applicable, we need one of the two detectable blocks of 'Zone-B' to be all zeros. Let us assume it is the leftmost one. This happens with probability $b(d;d,p) = p^d$.

Then we need more than half of the d bits of 'Zone-A' with value '0' (or less than one half with value '1'), which happens with probability $\sum_{k=0}^{\lfloor \frac{d-1}{2} \rfloor} b(k;d,p)$). We also need that one of the two detectable blocks of 'Zone-B' is all zeros (with probability p^d) and the other with at least one '1' bit to (which causes Rule 2 not to be applied), and happens with probability $1 - b(d;d,p) = 1 - p^d$. As this can happen twice, one with each of the blocks of 'Zone-B' forced to have all bits to '0', the total probability is $2 \cdot p^d(1-p^d) \sum_{k=0}^{\lfloor \frac{d-1}{2} \rfloor} b(k;d,p)$.

The same rule is also executed if both blocks of 'Zone-B' have all bits to '0' (with probability p^{2d}) and the number of '1' bits in 'Zone-A' is greater than 0 (to make Rule 2 not applicable) and less than one half of the block length d. The total probability is $p^{2d} \sum_{k=1}^{\lfloor \frac{d-1}{2} \rfloor} b(k;d,p)$.

b) In the case $b^1 = b^2$, the probability of decoding value '1' corresponds to the probability of decoding after applying Rule 1 or Rule 5 (note that Rule 3 is not applicable).

   - Rule 1 will be applied if all bits of 'Zone-A' are '1' and all bits of 'Zone-C' are '0'. In both cases, we need all bits to take the majority value, which happens with probability $b(2d;2d,p) = p^{2d}$.
   - The other possibility is to apply Rule 5 conditioned to not having applied Rule 1 before. There are two possible scenarios.

     In the first scenario, we need at least one bit of 'Zone-C' and more

than one half of the bits of 'Zone-A' with value '1'. This happens with probability $(1-p^d)\sum_{k=\lfloor\frac{d+2}{2}\rfloor}^{d} b(k;d,p)$.

In the other scenario, we need all bits of 'Zone-C' to be '0' and the number of ones in 'Zone-A' to be more than a half of the zone but less than d (otherwise Rule 1 would have been applied before).This happens with probability $p^d\sum_{k=\lfloor\frac{d+2}{2}\rfloor}^{d-1} b(k;d,p)$. □

Example: A possible collusion of three codewords of a $SC(4,3)$ code with $b^1 \neq b^2$ both encoding a '1' and b^3 encoding a '0'.

	Zone-A	Zone-B			Zone-C		
b^1	1111	1111	0000	0000	0000	0000	0000
b^2	1111	0000	1111	0000	0000	0000	0000
b^3	0000	0000	0000	0000	1111	0000	0000

Example: A possible collusion of three codewords of a $SC(4,3)$ code with $b^1 = b^2$ encoding a '1' and b^3 encoding a '0'.

	Zone-A	Zone-B			Zone-C		
b^1	1111	1111	0000	0000	0000	0000	0000
b^2	1111	1111	0000	0000	0000	0000	0000
b^3	0000	0000	0000	0000	1111	0000	0000

Figure 4.4 shows graphically the probability of decoding the majority value $p(v)$ as a function of the p-majority strategy applied over a $SC(d,t)$.

Table 4.7 present some numerical results on the lowest probability $p(v)$ of decoding the majority value v in a collusion of three buyers, for several scattering codes.

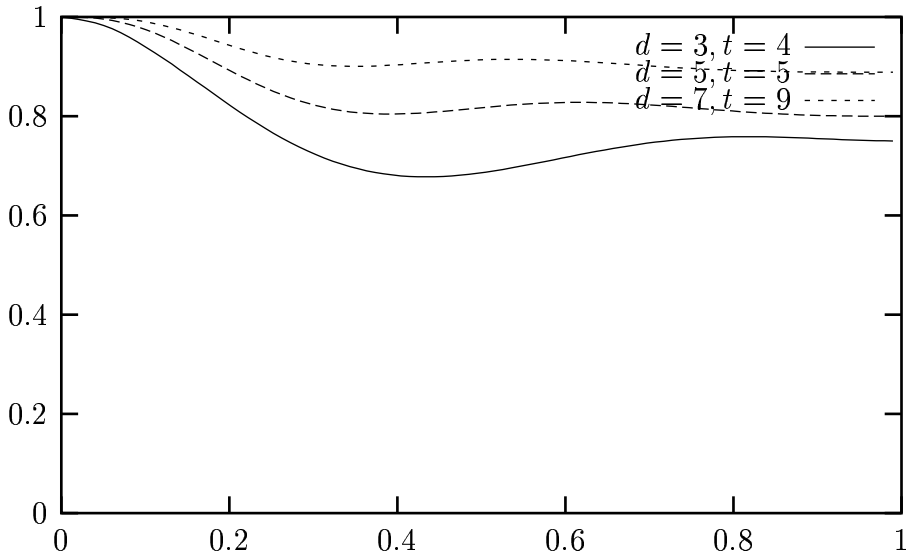

Figure 4.4: For different values of d and t, probability of decoding the majority value $p(v)$ as a function of the p-majority strategy applied over a $SC(d,t)$.

4.4 3-Secure codes

4.4.1 Construction

For $N = 2^n$ buyers, each buyer c^i is assigned a different codeword $a^i \in DH(n)$. Rather than directly embedding a^i in the content to be sold, the merchant generates a codeword A^i by composing a scattering code $SC(d,t)$ with a^i (See Figure 4.5). Such a composition is performed by replacing each bit of a^i with a codeword in $SC(d,t)$ which encodes the value of the bit of a^i. In this way, the codeword A^i will have bitlength

$$l = (N-1)(2t+1)d$$

d	t	$\min p(v)$
3	4	0.68
5	5	0.8
7	9	0.89
31	100	0.99

Table 4.7: Lowest probability $p(v)$ of decoding as the majority bit v in a collusion of three buyers, for several parameter choices (d, t).

The merchant then permutes the bits in A^i using a pseudo-random permutation seeded by a secret key known only to the merchant. The same permutation is applied to all codewords A^i. Figure 4.5 graphically depicts the construction described in this section. Finally, the merchant embeds the permuted version of A^i in the content being sold.

4.4.2 3-Collusions

Let us assume three dishonest buyers c^1, c^2, c^3 are assigned three codewords A^1, A^2, A^3 which have been built by:

1. Composing a scattering code with three different codewords $a^1, a^2, a^3 \in DH(n)$

2. Permuting the bits of the composed codewords

By comparison of their copies, the colluding dishonest buyers can identify $minor(A^1; A^2, A^3)$, $minor(A^2; A^1, A^3)$ and $minor(A^3; A^1, A^2)$. But as the bits of A^i have been secretly permuted, colluders cannot find out which bit of A^i corresponds to which bit of a^i. Thus, the colluders cannot identify $minor(a^1; a^2, a^3)$, $minor(a^2; a^1, a^3)$ nor $minor(a^3; a^1, a^2)$. Therefore, the only way for colluders to generate A^{coll} is to use a p-majority strategy.

According to Lema 9, all bits at positions $inv(a^1, a^2, a^3)$ remain unmodified after decoding each of the $2^n - 1$ components of A^{coll} to obtain a^{coll}. Also, according to Lema 10, all bits at positions $minor(a^i; a^j, a^k)$ for $(i, j, k) \in \{(1, 2, 3), (2, 1, 3), (3, 1, 2)\}$ will keep the majority value v (the one of a^j and a^k) with probability at least $p(v)$.

What is achieved with the above composition is that, regardless of the p-majority strategy used by colluders to generate words A^{coll}, the word a^{coll} resulting from decoding A^{coll} is a word generated by a $p(v)$-majority strategy collusion between a^1, a^2, a^3, where the value $p(v)$ is controlled by the merchant by choosing appropriate values for parameters d and t (see Table 4.7). It can be seen from Table 4.5 that controlling $p(v)$ is necessary to the keep low the probability ϵ of successful collusion. If A^i has some bits with value "unreadable", those bits are randomly set to '0' or '1'.

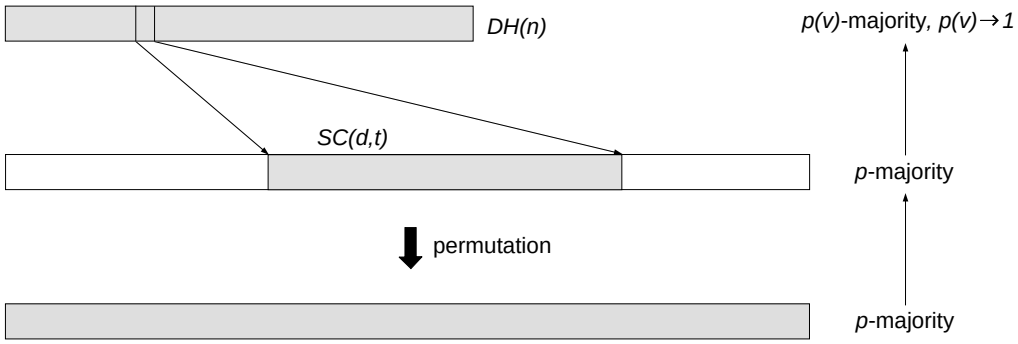

Figure 4.5: Construction of 3-secure codes.

4.4.3 Numerical results

Once parameters d and t have been fixed, the number of buyers can be increased by increasing n. For $d = 5$ and $t = 5$, Table 4.8 shows the size of the

code (number of buyers), the codeword length of our proposal, the probability of a successful collusion ϵ and the length of Boneh-Shaw's proposal for the same n and ϵ.

n	buyers	ϵ	Our length	Boneh-Shaw's length
7	128	$0.14 \cdot 10^{-6}$	6985	$2,788,320$
8	256	$0.15 \cdot 10^{-13}$	14025	$8,393,220$
9	512	$0.19 \cdot 10^{-27}$	28105	$28,340,928$

Table 4.8: Comparison between our codeword length and Boneh-Shaw's for the same number of buyers and security level (scattering code parameters: $d = 5$, $t = 5$)

It can be seen that Boneh-Shaw's construction results in much longer codewords than our proposal. Further, as n increases, their codeword length increases faster than ours.

In our proposal, once d and t have been fixed, the value ϵ decreases exponentially as n increases, which yields security levels higher than needed.

Thus, a better comparison is to use a fixed ϵ and assume that, for our security requirements, $\forall \epsilon' < \epsilon$ one has $\epsilon' \approx 0$. We take a value $\epsilon = 10^{-10}$ and use it as security level for Boneh-Shaw's construction. Results are presented in Table 4.9.

For a fixed $\epsilon = 10^{-10}$, we can observe that our proposal yields shorter codeword lengths up to $n = 16$ (number of buyers N is $65,536$). For values of $n > 16$ Boneh-Shaw's proposal offers a shorter codeword length. The explanation is that our codeword length increases as $O(N)$ while Boneh-Shaw's increases as $O(\log N)$ with a large constant factor; this large constant factor prevents Boneh-Shaw's scheme from comparing favorably unless N is very large.

buyers	Our length	Boneh-Shaw's length
512	$28,105$	$5,148,000$
$1,024$	$56,265$	$5,269,992$
$\ldots$	$\ldots$	$\ldots$
$32,768$	$1,802,185$	$5,883,888$
$65,536$	$3,604,425$	$6,006,780$
$131,072$	$7,208,905$	$6,129,816$

Table 4.9: Comparison of codeword length between our proposal and Boneh-Shaw's for the same number of buyers and assuming $\epsilon = 10^{-10}$ (scattering code parameters: $d = 5$, $t = 5$)