A note about the identifier parent property in Reed-Solomon codes

Marcel Fernandez a,*,1,3, Josep Cotrina a,2, Miguel Soriano a,b,1,3, Neus Domingo c

a Technical University of Catalonia, Department of Telematics Engineering, C/Jordi Girona 1 i 3, Mod C3, 08034 Barcelona, Spain
b CTTC, Centre Tecnologic de Telecomunicacions de Catalunya, Spain
c I.E.S. J.V. Foix, Barcelona, Spain

ARTICLE INFO

Article history:
Received 15 September 2009
Received in revised form
21 December 2009
Accepted 23 December 2009

Keywords:
Reed-Solomon codes
Fingerprinting
Traitor Tracing
Traceability codes
Identifiable parent property

ABSTRACT

Codes with traceability properties are used in schemes where the identification of users that illegally redistribute content is required. For any code with traceability properties, the Identifiable Parent Property (c-IPP) seems to be less restrictive than the Traceability (c-TA) property. In this paper, we show that for Reed-Solomon codes both properties are in many cases equivalent. More precisely, we show that for an \([n, k, d]\) Reed-Solomon code, defined over a field that contains the \(n - d\) roots of unity, both properties are equivalent. Also, we show how the strategy we propose can be applied to other cases by proving the equivalence of both properties for a particular code of characteristic 2. This answers a question posted by Silverberg et al. (2001, 2003), for a large family of Reed-Solomon codes.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

The concept of traitor tracing was coined in Chor et al. (1994) as a method to discourage piracy. Traitor tracing schemes are useful in scenarios where the distributed content may only be accessible to authorized users, like decrypting broadcast messages, software installation and distribution of multimedia content.

This paper discusses the characteristics of the identifiable parent property (IPP) of Reed-Solomon codes used in traitor tracing and fingerprinting schemes. However, before we get into technical matters, we give an intuitive overview. By doing this at the beginning of the paper, we try to separate the concepts from where our work emanates from the intrinsic mathematical development and also hopefully provide the reader an extra motivation for going deep into our results.

The scenario we will deal with is the following one. A distributor \(D\), that sells digital content, wishes to discourage illegal redistribution of its products. To this end, he embeds a unique set of symbols to each copy of the content before it is delivered. This makes each copy unique and therefore if a dishonest user illegally redistributes his copy, he can be unambiguously identified by simply extracting the set of symbols.

A weakness to this scheme can be spotted by noting that a coalition of two or more dishonest users can get together and by comparing their copies they perform a collusion attack. This attack consists in detecting the positions in which their copies differ and with this knowledge, they create a new copy...
that in every detected position contains a symbol of one of the 
members of the coalition. This new copy is a pirate copy that 
tries to disguise the identity of the guilty users and is the one 
they redistribute.

More precisely, the distributor assigns a codeword from a 
$q$-ary fingerprinting code to each user. To embed the code- 
word into each users object, the object is first divided into 
blocks. The distributor then picks a set of these blocks at 
random. This set of blocks is kept secret and will be the same 
for all users. Then using a watermarking algorithm a mark of 
the fingerprint codeword is embedded in each block. Note that 
a given user will have one of the $q$ versions of the block. The 
colluding traitors compare their copies, detect the blocks 
where their copies differ and with this information at hand, 
they construct a pirate copy where each block belongs to the 
corresponding block of one of the traitors. Since each mark is 
embedded using a different random sequence, and these 
sequences are unknown to the traitors, they cannot create a 
version of the block that they do not have.

With the above scenario in mind, it is clear that the 
distributor $D$, has to embed sets of symbols that are secure 
against collusion attacks. One way to obtain such sets is by 
using codes with the Identifiable Parent Property (c-IPP).

1.1. Previous work

Codes with the IPP were introduced in Hollmann et al. (1998). 
Informally, and using the traitor tracing scenario described 
above, a code has the c-IPP property if given a pirate copy, all 
coalitions of at most $c$ traitors that can generate this pirate 
copy have a non-empty intersection.

The IPP has received considerable attention in the recent 
years, having been studied by several authors (Barg et al., 2000; 
Barg and Kabatiansky, 2004; Tô and Safavi-Naini, 2004; Sarkar 
and Stinson, 2001; van Trung and Martirosyan, 2005; Alon 
et al., 2001; Alon and Stav, 2004; Fernandez and Soriano, 2002).

A stronger property is the Traceability (c-TA) property. In 
this case given a pirate copy, one of the traitors involved in its 
creation is the closest one in terms of the Hamming metric.

In Staddon et al. (2001), sufficient conditions for a linear 
error correcting code to be a c-TA code are given. Efficient 
algorithms for the identification of traitors in schemes using c-
TA codes are discussed in Silverberg et al. (2001, 2003).

In Silverberg et al. (2001, 2003) it is stated that tracing for TA 
codes is an $O(N)$ process, with $N$ the number of users, whereas 
for IPP codes tracing is more expensive since it is an $O(N^2)$ 
process. Since the TA property is stronger than the IPP, and 
tracing is far more expensive for the IPP, it seems natural to 
expect that by relaxing the TA requirements one could still 
have a code that, even though is no longer c-TA, still possesses 
IPP. However in Silverberg et al. (2003) some examples using 
truncated Reed-Solomon codes lead towards the opposite, 
that is, if a Reed-Solomon code does not have the TA property 
then it does not have the IPP one either.

1.2. Our contribution

In this paper we answer a question posted by Silverberg et al. 
(2001, 2003). The results we present hopefully give way to a 
total understanding of the IPP property in Reed-Solomon codes.

In Staddon et al. (2001), Lemma 1.3 authors prove that a c-
TA code is a c-IPP code. However as seen before, the TA 
property is stronger than IPP, taking this into account (Silver-
berg et al., 2001, 2003) asked the following question:

Question 11. Silverberg et al. (2003): It is the case that all c-IPP 
Reed-Solomon codes are c-TA codes?

Below, see also (Fernandez et al., 2009), and as a result of 
expressing the IPP in an algebraic manner, we give an affir-
mative answer to this question for a large family of Reed-
Solomon codes. Surprisingly enough, the answer is positive for 
codes defined over a field that contains the $n - d$ roots of unity. 
Note that our results imply that for this family of Reed-Solomon 
codes, failing to be c-TA also involves failing to be c-IPP.

Moreover, we show how the strategy adopted can be 
applied to other families of Reed-Solomon codes, namely to 
codes defined over fields of characteristic 2, by treating the 
perticular case of the field $F_{2^m}$. This is an extension of the work 
in Fernandez et al. (2009).

For a more precise statement of the Question 11 (Silverberg 
et al., 2003), see Section 2.1 below.

1.3. Organization of the paper

The paper is organized as follows. In Section 2, we provide the 
necessary background in coding theory, traceability and IPP. 
In Section 3 we start our discussion by defining a set of poly-
nomials that allow us to express the IPP algebraically. The 
main result of the paper is presented in Section 4, and comes 
in the form of a theorem giving the necessary and sufficient 
conditions for Reed-Solomon codes to be c-IPP codes. A 
complete example to clarify our results is given in Section 4.1. 
In Section 5 we show how our strategy is in a sense more 
general, and how can be applied to other families of Reed-
Solomon codes. We draw our conclusions in Section 6.

2. Definitions and previous results

We define a code as a set of $n$-tuples of elements from a set of 
scalars. The set of scalars is called the code alphabet. An $n$-tuple 
in the code is called a word and the elements of the code are 
called code words. If the code alphabet is a finite field $F_q$, then 
a code $C$ is a linear code if it forms a vectorial subspace. The 
dimension of the code is defined as the dimension of the 
vectorial subspace.

Let $a$, $b$ be $F_q^n$ be two words, then the Hamming distance 
$\text{d}(a, b)$ between $a$ and $b$ is the number of positions where $a$ 
and $b$ differ. Let $C$ be a code, the minimum distance of $C$, 
$d(C)$, is defined as the smallest distance between two different 
codewords.

A linear code with length $n$, dimension $k$ and minimum 
distance $d$ is denoted as a $(n, k, d)$-code, or simply as an $(n, d)$ 
code.

A well known class of linear codes are Reed-Solomon 
codes, that can be defined as follows:

Let $\mathbb{F}_q[x]$ be the ring of polynomials defined over $\mathbb{F}_q$. 
Consider the set of polynomials of degree less than $k$, 
$\mathbb{F}_q[x]_k \subseteq \mathbb{F}_q[x]$. Let $\gamma$ be a primitive element of $\mathbb{F}_q$, and 
$\lambda_0, \lambda_1, ..., \lambda_n \in \mathbb{F}_q - \{0\}$. 

\[ \text{C} = \{ \sum_{i=0}^{n} \lambda_i \gamma^i : \lambda_i \in \mathbb{F}_q \} \]
Definition 1. We define a generalized Reed-Solomon, RS[n, k, q], code as the vectorial subspace of $\mathbb{F}_q^n$ determined by the vectors of the form 
\[ \mathbf{v} = (\lambda_1 f(1), \ldots, \lambda_n f(n)) \]
where $f \in \mathbb{F}_q[x]$ and $\lambda_i \in \mathbb{F}_q$. Note that $n = q - 1$.

Definition 2. We define an extended Reed-Solomon, RS[n + 1, k, q], code as the vectorial subspace of $\mathbb{F}_q^{n+1}$ determined by the vectors of the form 
\[ \mathbf{v} = (\lambda f(0), \lambda_1 f(1), \ldots, \lambda_{n+1} f(n+1)) \]
where $f \in \mathbb{F}_q[x]$ and $\lambda_i \in \mathbb{F}_q$. Note that $n = q - 1$.

2.1. Background and previous results on c-IPP traceability codes

Given a code $C(n, d)$ defined over the finite field of $q$ elements, $\mathbb{F}_q$, where $n$ denotes the code length and $d$ the minimum distance of the code, the set of descendants (false fingerprint) of any subset $T = \{t^1, \ldots, t^\ell\} \subseteq C$, where $t^i = (t^i_1, \ldots, t^i_m)$, denoted $\text{desc}(T)$, is defined as
\[ \text{desc}(T) = \{y = (y_1, \ldots, y_m) \in \mathbb{F}_q^m | \exists t^i \in T, 1 \leq i \leq \ell \} \]

Definition 3. A code $C$ is a $c$-traceability code (denoted $c$-TA), for $c > 0$, if for all subsets (coalitions) $T \subseteq C$ of at most $c$ code words, if $y \notin \text{desc}(T)$, then there exists a $t \in T$ such that $d(y, t) < d(y, w)$ for all $w \in C - T$.

Definition 4. A code $C(n, d)$, defined over $\mathbb{F}_q$, is a c-identifiable parent property code (denoted $c$-IPP), $c > 0$, if for all $y \in \mathbb{F}_q^n$ and all the coalitions $T \subseteq C$ of at most $c$ code words, we have $y \notin \mathbb{V}_q \text{desc}(T)$ or $\bigcap_{y \in \text{desc}(T)} T \neq \emptyset$.

In (Staddon et al., 2001), Lemma 1.3 it is shown that a c-TA code is a c-IPP code. In (6) and (7) (Theorem 4.4.14) it is proved that any $C(n, d)$ code with $d > n - n/c^2$ is a c-TA code. Moreover, if $C(n, d)$ is a code defined over $\mathbb{F}_q$, in (Staddon et al., 2001, Lemma 1.6) the authors show that if $|C| > c \geq q$ then $C$ is not a c-IPP code.

Also, in (Silverberg et al., 2003, Theorem 8) the authors construct a family of truncated $(n < q - 1)$ RS[n, k, q] codes that fail to be c-IPP if $c^2 > n(n - d)$. Then in (Silverberg et al., 2003, Question 11) the authors ask if it is always true that the c-IPP fails if $c^2 > n(n - d)$.

In this paper we give an alternative positive answer of this question, showing that there are other families of Reed-Solomon codes that fail to be c-IPP if $c^2 > n(n - d)$. Obviously this does not close the problem, but we think that it gives some hints that may hopefully be useful in finding the final response.

3. The IPP condition for Reed-Solomon codes

In this section we set the ground for the discussion of our main results. Informally, we define a set of polynomials (denoted $h_i(x)$), that help us construct an algebraic representation of the IPP. Using these polynomials, we set up a system of equations for which the existence of a solution implies that the code is not c-IPP. In Section 4, we will show how to solve this equation system for a large number of Reed-Solomon codes.

Let $0 < c_1 \leq c_2$ be two integer numbers. We say that a code $C$ is not a $(c_1, c_2)$-IPP code if there exist coalitions $T_1$ of $c_1$ code words and $T_2$ of $c_2$ codewords, such that 
\[ \text{desc}(T_1) \cap \text{desc}(T_2) \neq \emptyset \quad \text{and} \quad T_1 \cap T_2 = \emptyset. \]

Obviously, from Definition 4 the code $C$ is not $c_2$-IPP.

Suppose that in a RS[n, k, q] code there exist two disjoint coalitions, $T_1$ and $T_2$, with $c_1$ and $c_2$ distinct code words respectively. We denote these coalitions as $T_1 = \{f_0(x), \ldots, f_{c_1-1}(x)\}$, $T_2 = \{g_0(x), \ldots, g_{c_2-1}(x)\}$ where $f(x), g(x) \in \mathbb{F}_q[x]$ but with an abuse of notation they also represent vectors of the form 
\[ f_i = (\lambda_0 f_i(0), \ldots, \lambda_{n-1} f_i(n-1)) \]
with $T_1 \cap T_2 = \emptyset$. If $T_1$ and $T_2$ can generate the same descendant (false fingerprint) $y$ then it is clear that the RS[n, k, q] code fails to be $(c_1, c_2)$-IPP.

We can always assume that code word $0$ is a code word of coalition $T_1$, otherwise consider coalitions $T_1 - f_0 = \{f_0 - f_0, \ldots, f_{c_1-1} - f_0\}$ and $T_2 - f_0 = \{g_0 - f_0, \ldots, g_{c_2-1} - f_0\}$. Then it is not difficult to verify that $(T_1 - f_0) \cap (T_2 - f_0) = \emptyset$ and they both can generate the fingerprint $y - f_0$. Thus, in what follows, we will assume that $f_0 = 0$.

We define polynomials 
\[ h_i(x) \triangleq f_i(x) - g_i(x) = \beta_i \prod_{k=1}^{c_2} (x - a_k^i) \]
for $i = 0, \ldots, c_1 - 1$ and $j = 0, \ldots, c_2 - 1$.

The polynomials $h_i(x)$ will be a key tool in all the subsequent work. In a sense, they allow us to have an algebraic representation of the IPP.

Note that the polynomials $h_i(x)$ have at most $n - d - k - 1$ roots, thus $s_i \leq n - d$, otherwise two distinct code words in the code would agree in more than $n - d$ coordinates, and this is not possible.

We will make an extensive use of the following result:

Lemma 5. If a RS[n, k, q] code fails to be $(c_1, c_2)$-IPP, $(T_1$ and $T_2$ can generate the same descendant), then the set of roots of the set of polynomials $\{h_i(x)\}$ contains $\mathbb{F}_q - \{0\}$. Therefore, $\sum_{i=0}^{c_1-1} s_i \geq n, x^n - 1 = \prod_{i=0}^{c_1-1} h_i(x)$ and $c_2(n - d) \geq n$.

Proof. The proof is straight forward from the definition of the polynomials $h_i(x)$ and the definition of the $(c_1, c_2)$-IPP.

In the previous reasoning we have seen that we always can take $f_0(x) = 0$, therefore 
\[ g_j(x) = f_j(x) - h_j(x) = -\beta_j \prod_{k=1}^{c_2} (x - a_k^j) \]
for $i = 0, \ldots, c_1 - 1$ then we can write down the following equation system (with an abuse of notation, because we are assuming that $s_i = n - d$ for all $i, j$, that in fact is the worst case situation):
\[
f_i^0 = \beta_0 \prod (-a_i^{k_j}) - \beta_0 \prod (-a_i^{k_j})
\]
\[
f_i^{n-d} = -\beta_0 \sum a_i^{k_j} + \beta_0 \sum a_i^{k_j}
\]
\[
f_i^{n-d} = \beta_i - \beta_0
\]

where \(i = 1, \ldots, c_i - 1\) and \(j = 0, \ldots, c_2 - 1\).

Note that if we assume that the values of \(a_i^{k_j}\) are known, we can ensure that one of the \(a_i^{k_j}\) codes corresponds to the coalition agrees with \(y\) in at least \(n - c_1(n - d)\) coordinates of \(y\), which is, that they can not generate the descendant \(y\).

For the necessary condition, in virtue of Lemma 2 we can assume that \(c_1c_2 = \lfloor n/(n - d) \rfloor \).

If \(n - d\) divides \(q - 1\), we have that the \((n - d)\)-roots of the unity belong to \(\mathbb{F}_q\). Let \(s = (q - 1)/(n - d)\), then we can express the \((n - d)\)-roots of the unity as \(\alpha^{ks}\), where \(\alpha\) is a primitive element of \(\mathbb{F}_q\).

4. Main result on IPP Reed-Solomon codes

4.1. Example

In this section we present an example of the above results.
We take a Reed-Solomon code over $\mathbb{F}_{13}$ ($q = 13$). We denote the elements of $\mathbb{F}_{13}$ as $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$. Since we wish to prove that if $d < n - (n/c^2)$, then the code is not c-IPP, we then take the code with parameters $[n = 12, k = 4, d = 9]$ and $c = 2$. Note that $n - d$ divides $q - 1$ ($\mathbb{F}_{13}$ contains the $n - d = 3$ roots of unity).

With the above reasoning in mind, we need to find polynomials $f_0(x), f_1(x), g_0(x), g_1(x)$ such that when grouped into two disjoint coalitions the corresponding code words can generate the same descendant (false fingerprint). In other words, we wish to find (disjoint) Coalition 1 $\{f_0(x) = 0, f_1(x)\}$ and Coalition 2 $\{g_0(x), g_1(x)\}$ such that their corresponding code words $[f_0, f_1]$ and $[g_0, g_1]$ can generate the same exact descendant (false fingerprint).

First of all, we define the $h_i$ polynomials.

$$
\begin{align*}
h_0 &= f_0 - g_0 = \beta \alpha_0^i (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) \\
h_1 &= f_1 - g_1 = \beta \alpha_1^i (x - \alpha_1^9)(x - \alpha_1^{10})(x - \alpha_1^{11}) \\
h_{10} &= f_1 - g_0 = \beta \alpha_0 (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) \\
h_{11} &= f_1 - g_1 = \beta \alpha_1 (x - \alpha_1^9)(x - \alpha_1^{10})(x - \alpha_1^{11}) \\
\end{align*}
$$

(7)

where the $\alpha_i^j$ take all of the non-zero values of $\mathbb{F}_{13}$.

Taking into account that $f_0(x) = 0$, we have that:

$$
\begin{align*}
g_0 &= -h_0 = -\beta \alpha_0 (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) \\
g_1 &= -h_1 = -\beta \alpha_1 (x - \alpha_1^9)(x - \alpha_1^{10})(x - \alpha_1^{11}) \\
f_1 &= h_{10} + g_0 = \beta \alpha_0 (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) - \beta \alpha_0 (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) \\
f_1 &= h_{11} + g_1 = \beta \alpha_1 (x - \alpha_1^9)(x - \alpha_1^{10})(x - \alpha_1^{11}) - \beta \alpha_0 (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) \\
\end{align*}
$$

(8)

and since $f_1(x) = f_1^2 + f_1(x) + f_1^2(x)$ (because $k - 1 = n - d = 3$), it follows that the system to be solved is

$$
\begin{align*}
f_1^2 &= -\beta \alpha_0 (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) + \beta \alpha_0 (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) \\
f_1^2 &= \beta \alpha_1 (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) + \beta \alpha_0 (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) \\
f_1^2 &= -\beta \alpha_0 (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) + \beta \alpha_0 (x - \alpha_0^9)(x - \alpha_0^{10})(x - \alpha_0^{11}) \\
\end{align*}
$$

(9)

Next, we define the $h_i(x)$ polynomials as

$$
h_i(x) = \gamma_i x^{(i+1)(n-d)}x^{n-d} - \gamma_i
$$

(10)

with $i = \{0, 1\}$, $j = \{0, 1\}$ the integer value $c = 2$ and $\alpha = 2$ a primitive element of $\mathbb{F}_{13}$, so

$$
h_i(x) = \gamma_i x^{(i+1)(n-d)}x^{n-d} - \gamma_i
$$

(11)

Note that according to (3) we have that,

$$
\beta_i = \gamma_i x^{(i+1)(n-d)}
$$

Now by plugging (11) in (8), the equation system (9) becomes:

$$
\begin{align*}
(i = 1, j = 0) & \quad f_1^2 = -\gamma_{10} + \gamma_{00} \\
(i = 1, j = 1) & \quad f_1^2 = -\gamma_{11} + \gamma_{10} + \gamma_{01} \\
\end{align*}
$$

(12)

$$
\begin{align*}
(i = 1, j = 0) & \quad f_1^2 = -\gamma_{10} + \gamma_{00} \\
(i = 1, j = 1) & \quad f_1^2 = 2\gamma_{11} + \gamma_{01} \\
\end{align*}
$$

(13)

We take for instance (12) (taking (13) leads to a similar result). As seen in (5), we have that:

$$
f_1^2 = 0
$$

now taking $\gamma_{00} = 1$ and using (6), yields

$$
\begin{align*}
(i = 1, j = 0) & \quad \gamma_{00} = 1 \\
(i = 1, j = 1) & \quad \gamma_{11} = 2
\end{align*}
$$

(14)

Therefore,

$$
f_0(x) = 0 \\
f_1(x) = 2
$$

(15)

Using these values in (13):

$$
\begin{align*}
(i = 1, j = 1) & \quad 2 = -\gamma_{11} + \gamma_{01} \\
0 & \quad = 2^2\gamma_{11} - 2\gamma_{01}
\end{align*}
$$

(16)

solving, we have that

$$
\gamma_{01} = 1 \quad \text{and} \quad \beta_{11} = 12
$$

(17)

Which yields

$$
\begin{align*}
h_{10}(x) &= x^3 - 1 = x^3 + 12 \\
h_{11}(x) &= 2^2x^3 - 1 = 8x^3 + 12 \\
h_{10}(x) &= 12 - 2^2x^3 - 12 = x^3 + 1 \\
h_{11}(x) &= 12 - 2^2x^3 - 12 = 8x^3 + 1 \\
\end{align*}
$$

(18)

Finally, using (8) we have

$$
\begin{align*}
g_0(x) &= 12x^3 + 1 \\
g_0(x) &= 5x^3 + 1
\end{align*}
$$

(19)

We have arrived at Coalition 1: $f_0(x) = 0, f_1(x) = 2$ and Coalition 2: $g_0(x) = 12x^3 + 1, g_1(x) = 5x^3 + 1$.

Encoding these polynomials, we have that for Coalition 1:

$$
\begin{align*}
f_0 &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\
f_1 &= (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)
\end{align*}
$$

(20)

and for Coalition 2:

$$
\begin{align*}
g_0 &= (6, 0, 6, 2, 6, 9, 0, 2, 9, 2) \\
g_1 &= (6, 2, 6, 9, 0, 2, 9, 0, 0, 6, 9, 0)
\end{align*}
$$

(21)

It is clear that both coalitions can create the same descendant (false fingerprint):

$$
(0, 2, 0, 2, 2, 0, 0, 0, 0, 2, 0, 2)
$$

(22)

5. The case of fields of characteristic 2 and $c = 2$ colluders

In the previous section, we proved that for a $[n, k, d]$ Reed-Solomon code defined over the field $\mathbb{F}_{q}$, if $n - d$ divides $q - 1$, then the code is c-IPP if and only if $c^2 < n/(n - d)$. Therefore, at first glance it seems that the use of the $h_i$ polynomials defined in (1) only applies to the case where $n - d$ divides $q - 1$. In this
section we show that this need not be the case, and that the strategy of using the polynomials $h_i$, can be applied to a wider range of codes. For our discussion we use the $[n = 16, k = 5, d = 12]$ Reed-Solomon code defined over $\mathbb{F}_2^5$. Note that according to Definition 2 this is an extended Reed-Solomon code. It is clear that for this code $d \leq n - (n/c^2)$ if $c = 2$, and we wish to prove that the code is not 2-IPP. Note that this is a worst case situation, since for a larger distance the code is 2-IPP.

To start with, suppose that in the $[n = 16, k = 5, d = 12]$ Reed-Solomon there is a codeword of the form

$$h = \left\{ (0, 0, 0, 0, a, a, a, a, a, a, a, a, a) \right\}$$

with $a + \beta = \gamma$. Then, by the linearity of the code and since the underlying field is of characteristic 2, we also have codewords

$$h_{02} = (a, a, a, a, 0, 0, 0, 0, 0, 0, 0, 0)$$

and

$$h_{10} = (0, 0, 0, 0, a, a, a, a, a, a, a, a)$$

It is important to note that the polynomials $h_{00}(x), h_{01}(x), h_{10}(x)$ and $h_{11}(x)$, associated with codewords $h_{00}, h_{01}, h_{10}$ and $h_{11}$ respectively, have as zeros all the elements of the field $\mathbb{F}_2$, and therefore they are equivalent to the polynomials in (1).

Now, using again (1), and taking $f_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, we have that

$$h_{00} = f_0 - g_0 = g_0 = (0, 0, 0, 0, a, a, a, a, a, a, a, a, a)$$

and

$$h_{01} = f_0 - g_1 = g_1 = (a, a, a, a, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

For the coalition $\{g_0, g_1\}$,

$$g_0 = (0, 0, 0, 0, a, a, a, a, a, a, a, a, a)$$

and

$$g_1 = (a, a, a, a, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

can create the same descendant:

$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

Note that the key issue in the above reasoning is (20), so we need a codeword whose positions hold only 4 different symbols, in other words, in the polynomial associated with the codeword:

$$p(x) = h_{00}(x) = (x + a_0)(x + a_2)(x + a_3)(x + a_5)$$

the elements $a_0, a_2, a_3, a_5$ need to be such ones that when evaluated in all the elements of the field, the range of $p(x)$ consists in exactly 4 different elements.

Observe that the above reasoning shows that the $[n = 16, k = 5, d = 12]$ Reed-Solomon code is not 2-IPP, subject to the fact of obtaining a polynomial $p(x)$ with an associated codeword containing exactly 4 different elements.

More precisely, we need to find 4 disjoint sets of elements in $\mathbb{F}_2$, say,

$$\mathbf{a}_1 = \{ a_{11}, a_{12}, a_{13}, a_{14} \}$$

$$\mathbf{a}_2 = \{ a_{21}, a_{22}, a_{23}, a_{24} \}$$

$$\mathbf{a}_3 = \{ a_{31}, a_{32}, a_{33}, a_{34} \}$$

$$\mathbf{a}_4 = \{ a_{41}, a_{42}, a_{43}, a_{44} \}$$

such that

$$p(a_{11}) = p(a_{12}) = p(a_{13}) = p(a_{14})$$

$$p(a_{21}) = p(a_{22}) = p(a_{23}) = p(a_{24})$$

$$p(a_{31}) = p(a_{32}) = p(a_{33}) = p(a_{34})$$

$$p(a_{41}) = p(a_{42}) = p(a_{43}) = p(a_{44})$$

To begin with, we take any $a_{11} \notin \{a_0, a_2, a_3, a_5\}$. Then, we have that

$$a_{11} + a_0 = a_{61}$$

$$a_{11} + a_2 = a_{62}$$

$$a_{11} + a_3 = a_{63}$$

$$a_{11} + a_5 = a_{64}$$

with

$$p(a_{11}) = a_{61}, a_{62}, a_{63}, a_{64}$$

for the appropriate $\{a_{61}, a_{62}, a_{63}, a_{64}\}$.

Since according to (27), we want that $p(a_{11}) = p(a_{12})$, we now choose a value $a_{12}$ such that

$$a_{12} + a_0 = a_{41}$$

It is clear that such a value always exists. Also, from (28), we note that

$$a_{11} + a_d = a_{12} + a_0 = a_{64}$$

and also that

$$a_d \neq a_0 \Rightarrow a_{11} \neq a_{12}$$

It follows that

$$a_{12} + a_0 = a_{64}$$

where the first equality is obtained by using (29) and the second by noting from (28) that $a_c + a_d = a_{613} + a_{614}$. By proceeding similarly, we arrive at

$$a_{613} + a_0 = a_{614}$$

$$a_{612} + a_0 = a_{614}$$

and that

$$a_0 \neq a_2 \Rightarrow a_{13} \neq a_{12}$$

Again, it is clear that such a value always exists. As before, from (28) and (30) we have that

$$a_{13} + a_0 = a_{12} + a_0 \text{ such that } a_0 \neq a_{12} \Rightarrow a_{13} \neq a_{12}$$

or

$$a_0 \neq a_{12} \Rightarrow a_{13} \neq a_{12}$$

Following this line of reasoning, it is easy to see that we can also always find an element $a_{14} \notin \{a_{11}, a_{12}, a_{13}\}$ such that
Therefore, we can write
\[ p(a_{11}) = p(a_{12}) = p(a_{13}) = p(a_{14}) = a_{81}a_{82}a_{83}a_{84} \]

(35)

Moreover, using (31) we also have that
\[ a_4 + a_8 + a_{12} + a_{16} = 0 \]
\[ a_{11} + a_{12} + a_{13} + a_{14} = 0 \]
\[ a_{41} + a_{42} + a_{43} + a_{44} = 0 \]

(36)

(37)

At this stage, we take \( \beta \notin \{a_{11}, a_{12}, a_{13}, a_{14}\} \). From (28) we have
\[ a_{11} + \beta + a_3 = a_{81} + \beta \]
\[ a_{11} + \beta + a_9 = a_{82} + \beta \]
\[ a_{11} + a_3 + a_9 + a_{14} + \beta = 0 \]
and
\[ p(a_{11} + \beta) = (a_{81} + \beta)(a_{82} + \beta)(a_{83} + \beta)(a_{84} + \beta) \]

(38)

We proceed by using (30), (34) and (35) to express (36) as
\[ p(a_{11} + \beta) = p(a_{12} + \beta) = p(a_{13} + \beta) = p(a_{14} + \beta) \]
\[ = (a_{81} + \beta)(a_{82} + \beta)(a_{83} + \beta)(a_{84} + \beta) \]

(39)

The following definitions
\[ a_{11} + \beta \oplus a_{21}, \quad a_{11} + \beta \oplus a_{22}, \]
\[ a_{12} + \beta \oplus a_{22}, \quad a_{13} + \beta \oplus a_{23}, \]
\[ a_{14} + \beta \oplus a_{24}, \quad a_{11} + \beta \oplus a_{24}, \]
\[ a_{12} + \beta \oplus a_{23}, \quad a_{13} + \beta \oplus a_{22}, \]
\[ a_{14} + \beta \oplus a_{21}, \quad a_{11} + \beta \oplus a_{21}, \]

allow us to express (39) as:
\[ p(a_{21}) = p(a_{22}) = p(a_{23}) = p(a_{24}) = a_{81}a_{82}a_{83}a_{84} \]

(40)

(41)

with
\[ a_{21} + a_{22} + a_{23} + a_{24} = 0 \]

(42)

It is intuitively clear that we can proceed accordingly, and obtain the following expressions
\[ a_{21} + a_{22} + a_{23} + a_{24} = 0 \]
\[ p(a_{31}) = p(a_{32}) = p(a_{33}) = p(a_{34}) = a_{81}a_{82}a_{83}a_{84} \]
\[ a_{41} + a_{42} + a_{43} + a_{44} = 0 \]
\[ p(a_{41}) = p(a_{42}) = p(a_{43}) = p(a_{44}) = a_{81}a_{82}a_{83}a_{84} \]

(43)

Therefore, with (43) in mind and according to (26) and (27), if we are able to group the elements of the field as
\[ 0 + a_{12} + a_{13} + a_{14} = 0 \]
\[ a_{21} + a_{22} + a_{23} + a_{24} = 0 \]
\[ a_{31} + a_{32} + a_{33} + a_{34} = 0 \]
\[ a_{41} + a_{42} + a_{43} + a_{44} = 0 \]

then the elements in the codeword \( h_{00} \), associated with polynomial
\[ p(x) = h_{00}(x) = x(x - a_{12})(x - a_{13})(x - a_{14}) \]
will consist in only 4 different symbols. The following lemma solves the issue.

Lemma 8. The field \( F_{16} \) has an additive subgroup of 4 elements, say \( \{0, a, \beta, \gamma\} \), with \( a + \beta + \gamma = 0 \). Moreover, the sum of the elements of each coset of this subgroup is 0.

Proof. By Lagrange’s Theorem (Artin, 1991) the order of a subgroup divides the order of the group it belongs to, so it is possible to have an additive subgroup of order 4 in \( F_{16} \). Note that 0, is an element of the subgroup. Since the field is of characteristic 2, every element is its own additive inverse. Also by construction, since \( a + \beta + \gamma = 0 \), the sum of any two elements of the subgroup belongs to the subgroup.

Now, we take \( c(F_{16}) \), with \( c \notin \{0, a, \beta, \gamma\} \) and let \( c_1 = c + \{0, a, \beta, \gamma\} \) be the associated coset. Observe that \( c + e \in \{0, a, \beta, \gamma\} \) is in fact true. That is, all \( c \) Reed-Solomon codes defined over a field that contains the elements \( h(x) \), also allows us to obtain the required information about the IPP property.

6. Conclusions

In this paper we have discussed the IPP in Reed-Solomon codes. The goal of our work was to answer a question by Silverberg et al. (2001, 2003), inquiring whether all c-IPP Reed-Solomon codes are also c-TA codes. We have developed a strategy that consists in expressing the IPP algebraically through the definition of a suitable set of polynomials. Using these polynomials, we are able to show that for a large family of Reed-Solomon codes this is in fact true. That is, all \( n, k, d \) Reed-Solomon codes defined over a field that contains the \( n - d \) roots of unity are IPP codes if and only if they are also TA codes. In addition, we show that our discussion can be applied to Reed-Solomon codes defined over fields of characteristic 2. More precisely, we show that the \( n = 16, k = 5, d = 12 \) Reed-Solomon code defined over \( F_{16} \) is not 2-IPP. We stress that the extension to other codes is non-trivial.

It is surprising that from our results it seems that the IPP characteristics of a Reed-Solomon code lie solely in the field over which the code is defined. To devise the exact extension of this dependence will be a subject of further research.

**References**


Marcel Fernandez is an associate professor at the Universitat Politècnica de Catalunya, Barcelona, Spain. His research interests are in coding theory and its applications in digital content protection.

Josep Cotrina is a professor at the Universitat Politècnica de Catalunya, Barcelona, Catalunya. His research interests are in educational bubble, coding theory and its applications in digital content protection.

Miguel Soriano received the Telecommunication Engineering degree and the Ph.D. from the Technical University of Catalonia (UPC), Barcelona, Spain. In 1991, he joined the Cryptography and Network Security Group at the Department of Applied Mathematics and Telematics. Now, he works with the Information Security Workgroup within the Telematics Services Research Group at the Department of Telematics Engineering of the UPC. Since 2007 he is Professor at the UPC, where he teaches and coordinates undergraduate and graduate courses in Data Transmission, Cryptography and Network Security and E-commerce. His current research interests include information and network security including information hiding for copyright protection.

Neus Domingo is a professor at the Pla Farreres School, Barcelona, Catalunya. His research interests are in coding theory.