Holzbaur, Lukas; Freij-Hollanti, Ragnar; Wachter-Zeh, Antonia; Hollanti, Camilla

Private streaming with convolutional codes

Published in:
2018 IEEE Information Theory Workshop, ITW 2018

DOI:
10.1109/ITW.2018.8613505

Published: 15/01/2019

Please cite the original version:
Private Streaming with Convolutional Codes

Lukas Holzbaur*, Ragnar Freij-Hollanti*, Antonia Wachter-Zeh*, Camilla Hollanti†

*Institute for Communications Engineering, Technical University of Munich, Germany
†Department of Mathematics and Systems Analysis, Aalto University, Finland

Email: {lukas.holzbaur, ragnar.freij, antonia.wachter-zeh}@tum.de
Email: camilla.hollanti@aalto.fi

Abstract—Recently, information-theoretic private information retrieval (PIR) from coded storage systems has gained a lot of attention, and a general star product PIR scheme was proposed. In this paper, the star product scheme is adopted, with appropriate modifications, to the case of private (e.g., video) streaming. It is assumed that the files to be streamed are stored on \( n \) servers in a coded form, and the streaming is carried out via a convolutional code. The star product is defined for this special case, and various properties are analyzed in the baseline case, with colluding servers, as well as with straggling and byzantine servers. The achievable PIR rates are derived for the given models.

I. INTRODUCTION

Private information retrieval (PIR) studies the problem when a user wants to retrieve a file from a storage system without revealing the identity of the file in question to the storage servers. The original problem was introduced in [1], [2], and more recently the problem setting was extended to the case where the files are stored on the servers in an encoded form rather than merely being replicated [3]–[5]. The capacity of PIR with and without colluding and Byzantine servers was derived in [6]–[9]. In [10], a so-called star product PIR scheme was introduced. The scheme works with any linear code as a storage code and retrieval code, and achieves the highest rate when both codes are generalized Reed-Solomon codes.

Currently Netflix and Youtube alone are occupying more than 50% of Internet downstream traffic. Motivated by this huge increase in multi-media streaming, we will consider private streaming suitable for distributed systems sharing encoded streams. In a wider context, this is related to the problem of private stream search (PSS), which has been considered, e.g., in [11]–[13], typically using cryptographic assumptions. In this paper, we require information-theoretic privacy, namely that the servers gain zero information on the index of the file requested for streaming, based on the received query.

Since convolutional codes are suitable for streaming and the star product scheme is efficient and flexible for PIR [14], [15], we will design a scheme that combines these approaches.

Convolutional codes are sensitive to burst errors but good at handling well-distributed errors. As burst errors are unlikely on, e.g., an additive white Gaussian noise (AWGN) channel, they exhibit good performance and have a lower bit error rate on such channels than comparable block codes with the same rate. See [16, Section V] for more details.

The main contributions of this paper are the following.

- To the best of the authors’ knowledge, information-theoretically private streaming is considered for the first time.
- Memory is introduced into the star product PIR scheme by a block convolutional structure, improving the performance of the decoder for a large class of channels.
- Two schemes for different channels, namely a block erasure channel and a non-bursty channel, e.g., an AWGN channel, are given. Both can operate on the same database and the user can adapt the queries according to the current channel conditions.

II. PRELIMINARIES

We denote by \([n,b]\) the set of integers \(\{i \mid a \leq i \leq b\}\) and \([b] = [1,b]\). If \(c\) and \(d\) are vectors of the same length \(n\), we define their star product as the coordinate-wise product

\[c \star d = (c_1d_1, \ldots, c_n d_n).\]

Further, if \(C\) and \(D\) are linear codes of same length, we define their star product to be the linear code given by the span

\[C \star D = \langle c \star d \mid c \in C, d \in D \rangle.\]

Throughout the paper, \(\mathbb{F}\) will denote an arbitrary finite field.

A. Convolutional Codes

Definition 1 (Convolutional code). Let \(G_1, \ldots, G_{M+1} \in \mathbb{F}^{k \times n}\) and \(\text{rank}(G_1) = k\). Define an \((n,k)\) convolutional code \(C_c\) as

\[Y_i = \sum_{j=1}^{M+1} X_{i-j+1} G_j, \quad (1)\]

where \(X_0, \ldots, X_{M+1} = 0\) and \(X_j \in \mathbb{F}^k\).

We refer to \(M\) as the memory of \(C_c\), and if \(M = 1\), we say that \(C_c\) is a unit memory (UM) code.

We consider terminated convolutional codes, i.e., \(Y\) is not a semi-infinite vector, but \(Y = (Y_1, Y_2, \ldots, Y_{t+M})\) with \(Y_t\) as in (1).

An \((n,k)\)-code denotes a linear block code of length \(n\) and dimension \(k\). A generalized Reed–Solomon (GRS) code
\(\mathcal{RS}(n,k,v)\) is an \((n,k)\)-code with minimum distance \(d = n - k + 1\) and generator matrix

\[
G = \begin{pmatrix}
1 & 1 & \cdots & 1 \\
\alpha_1 & \alpha_2 & \cdots & \alpha_n \\
\vdots & \vdots & \ddots & \vdots \\
\alpha_{k-1} & \alpha_{k-2} & \cdots & \alpha_{n-k+1}
\end{pmatrix}
\begin{pmatrix}
v_1 \\
v_2 \\
\vdots \\
v_n
\end{pmatrix},
\]

where \(\alpha_1, \ldots, \alpha_n \in \mathbb{F}\) are distinct evaluation points and the \(v_j\)'s are all non-zero. If the choice of the \(v_j\)'s is not important, we sometimes write \(\mathcal{RS}(n,k)\) code.

The distance measure of interest for convolutional codes is the extended row distance \(d^e\), which determines the minimum number of errors required for an error burst of \(t\) blocks to occur. For UM codes this distance can be lower bounded by the designed extended row distance

\[
d^e = d_1 + (t - 1)d_2 + d_2, \quad t \geq 1,
\]

where \(d_1, d_2,\) and \(d_3\) denote the distances of the codes generated by \(G_1, G_2,\) and \([G_1^T, G_2^T]^T\) respectively. In [16], a decoding algorithm is given, which is guaranteed to be successful if the number of errors does not exceed half the designed extended row distance for any \(t\), i.e.,

\[
\sum_{j=s}^{i+s} W_H(w_j) < \frac{d^e}{2}, \quad \forall \ s \in [\ell + M], t \in [0, \ell + M - s], \tag{3}
\]

where \(w_j\) denotes the error vector of the \(j\)-th block. We refer to an \((n,k)\) UM code for which \(d_3, d_1\) and \(d_2\) achieve the Singleton bound for block codes as an optimal \((n,k)\) UM code.

### B. Star Product PIR

We review the star product scheme for PIR from an arbitrary storage code, as introduced in [10]. Let \(C\) be an \((n,k)\) code (the storage code) with generator matrix \(G \in \mathbb{F}^{k \times n}\), storing \(m\) files \(X^1, \ldots, X^m \in \mathbb{F}^k\). This means that each server \(j \in [n]\) stores a column \(Y_j\) of the matrix \(Y = XG \in \mathbb{F}^{m \times n}\), where \(X \in \mathbb{F}^{m \times k}\) is a data matrix, whose \(i\)-th row \(X_i\) represents the \(i\)-th file. The scheme we will describe allows a user to retrieve the file \(X_i\) without disclosing the index \(i\).

Let \(D\) be a code of the same length as \(C\). Let \(D \in \mathbb{F}^{m \times n}\) be a matrix whose \(m\) rows are i.i.d. uniformly random codewords of \(D\). The query for the \(j\)-th server is given by

\[
q^j = D_{\cdot,j} + e_i E_{\cdot,j},
\]

where \(e_i\) denotes the \(i\)-th standard basis vector and \(E = E_1, E_2, \ldots, E_n\) is a vector\(^1\).

The servers now respond with the standard inner product of their \((m \times 1)\) stored vector \(Y_j\) and the query vector \(q^j\) which they received, so the query of the \(j\)-th server is the symbol

\[
r^j = \langle q^j, Y_j \rangle = \sum_{s=1}^{m} D_{s,j} Y^s_j + E_{1,j} Y^j_j \in \mathbb{F}. \tag{5}
\]

\(^1\)This notation is chosen to be consistent with the later sections when \(E\) will be a matrix.

Considering the \(n\) responses obtained as a vector in \(\mathbb{F}^{1 \times n}\), we can write it as

\[
r^j = \sum_{s=1}^{m} (D_{s,1}Y^s_1, \ldots, D_{s,n}Y^s_n) + (E_{1,1}Y^1_1, \ldots, E_{1,n}Y^1_n)
\in C \ast D + E \ast Y^1_i.
\]

Assuming \(E\) has weight \(W_H(E) < d_{C,D}\), erasure decoding in \(C \ast D\) now allows us to retrieve the vector \(E \ast Y^1_i\), which depends only on the desired file \(Y^1_i\). The rate achieved by this scheme is given by

\[
R_{PIR}^n = \frac{n - (k + t - 1)}{n} \tag{6}
\]

where \(t\) is the number of colluding servers, \(i.e.,\) the maximal number of servers that can exchange their queries such that the scheme is still private (see [10] for details). If the response is corrupted by channel erasures or a bounded number of malicious servers, the user first decodes the response in \(C \ast (D + E)\), as in [15]. This is discussed further in Section V.

### III. PIR FROM CONVOLUTIONAL CODES

In this section, it is shown how a large file can be streamed with asymptotically no rate loss by designing, as per user's request, the retrieved symbols such that they are codewords of a convolutional code.

#### A. Storage Code

In this work we only consider Reed-Solomon codes for the storage code as they are a well suited choice for both the star product scheme and block convolutional codes. Denote by \(m\) the number of files \(X^s \in \mathbb{F}^k, s \in [m]\), and by \(n\) the number of servers. The files are split into \(\ell\) stripes \(X^s_i \in \mathbb{F}^k\) and encoded with an \(\mathcal{RS}(n,k)\) storage code \(C\) with evaluation points \(\alpha_j, j \in [n]\). The \(j\)-th server stores the \(j\)-th symbol of every encoded stripe \(Y^s_i \in \mathbb{F}^n\) (see Figure 1).

#### B. Query

We query for a linear combination of \(M + 1\) stripes in each block and design the queries such that the responses are codewords of a convolutional code. Let \(D\) be an \(\mathcal{RS}(n, t)\) code; the matrix \(D \in \mathbb{F}^{(M+1)\times m}\) as in (4); and \(J \subset [n]\) with \(|J| \leq d_{C,D} - 1\). The query for the \(j\)-th server is given by

\[
q^j = \sum_{z=0}^{M} e_z \alpha^j_z, \quad z \in [0, M],
\]

where \(e_z\) is the \(z\)-th standard basis vector and the matrix \(E \in \mathbb{F}^{(M+1)\times n}\) is given by

\[
E_{z+1,j} = \begin{cases} \alpha^j_z, & \text{if } j \in J \\ 0, & \text{otherwise} \end{cases}
\]

#### C. Response

The protocol consists of \(\ell + M\) iterations in each of which the servers respond with the inner product of the vector and a vector containing the stored symbols of \(M + 1\) stripes of each file, depending on the iteration. In iteration \(\xi\) the response of server \(j\) is given by

\[
r^j_{\xi,j} = \langle q^j_{\xi,j}, [Y_{\xi,j}, Y_{\xi-1,j}, \ldots, Y_{\xi-M,j}]^T \rangle,
\]

\(^2\)Note that \(E_{z+1,j}\) is a scalar.
The highest PIR rate in this setting is achieved for $G$. Fig. 1. The queried symbols in iteration $\Unif(D)$ is given by

$$r_k^i = \sum_{s=0}^{M} \sum_{z=0}^{M} Y_{r_{i+1}}^s \cdot D_{2m+1} \cdot E_{i+1},$$  \hspace{1cm} (10)$$

An illustration of the responses for UM codes is given in Figure 2.

**Lemma 1.** Let $|J| \geq k$. Given the the responses $\{r_1^i, r_2^i, \ldots, r_k^i\}^i$ the file $X^i$ can be recovered.

**Proof:** By (8) the vectors $E_{i+1}$ are designed such that for any $c \in \mathcal{C} \cdot E_{i+1}$, $z \in [0, M]$ it holds that $c_z = 0$, $\forall j \notin J$. As $|J| \leq |C \cdot D|$ recovers the vector $\sum_{z=0}^{M} E_{i+1}^z \cdot Y_{r_i}^z = \sum_{z=0}^{M} X_{\xi - z}^i \cdot G_{\mathcal{C} \cdot E_{i+1}}$ in each iteration, where the $G_{\mathcal{C} \cdot E_{i+1}}$ are generator matrices of the storage code $\mathcal{C}$ with column multipliers $E_{i+1}$. Since $|J| \geq k$, each $G_{\mathcal{C} \cdot E_{i+1}}$ is of rank $k$ and it follows that given the set $\{X_{\ell - k}^i, \ldots, X_1^i\}$ of the stripe $X_1$ can be determined uniquely. In the first iteration $X_1 = \cdots = X_0 = 0$ so $X_1$ can be recovered and recovery of the remaining stripes follows by induction.

As both $\mathcal{C}$ and $\mathcal{D}$ are GRS, the distance of the star product $\mathcal{C} \cdot \mathcal{D}$ is given by $d_{\mathcal{C} \cdot \mathcal{D}} = n - (k + t - 1) + 1$ and it follows that at most $d_{\mathcal{C} \cdot \mathcal{D}} - 1 = n - k + t - 1$ symbols can be downloaded in each iteration. The PIR rate is given by

$$R = \frac{(n - (k + t - 1))}{(t + 1)n},$$  \hspace{1cm} (11)$$

which approaches the PIR rate of (10) for $t \to \infty$. The highest PIR rate in this setting is achieved for $|J| = k$ and $n = 2k + t - 1$.

IV. PROTECTING AGAINST BLOCK ERASURES

In the previous section, we showed how to design queries such that the symbols of the desired file recovered from the responses are symbols of a code of higher dimension and memory $M$. While this setting asymptotically achieves the same PIR rate as a comparable system that downloads blocks without memory, it offers no immediate advantages. In this section, we utilize the construction to design a PIR scheme that is able to stream files consisting of many stripes in the presence of burst block erasures, i.e., iterations where all the responses of the servers are lost. Since we are interested in streaming applications, decoding should be possible without a big delay and without querying for more data or retransmission of blocks. Therefore, we consider a sliding decoding window of $N$ blocks and denote the maximum burst length of block erasures in a window by $\epsilon$. To protect against these erasures, more symbols than in the setting of the previous section of each block have to be retrieved privately in each iteration.

**Lemma 2.** The number of symbols privately retrieved in each non-erased block has to satisfy $k \geq \frac{NK}{N-\epsilon}$.

**Proof:** The file $\mathcal{C}$ is of rank $k$ and there is only one symbol of $\mathcal{C}$ that belongs to each stripe $X_1$. Thus, there are $N$ symbols of $\mathcal{C}$ that are retrieved. To protect against the presence of burst block erasures, decoding should be possible without a big delay and without querying for more data or retransmission of blocks. Therefore, we consider a sliding decoding window of $N$ blocks and denote the maximum burst length of block erasures in a window by $\epsilon$. To protect against these erasures, more symbols than in the setting of the previous section of each block have to be retrieved privately in each iteration.

**Theorem 1.** Let $J$ be a set with the recovering property as in Definition 2 and $n \geq k + t - 1 + |J|$. For any set $\mathcal{R} = [\xi - N + \epsilon + 1, \ldots, \xi] \in [\ell + M]$; the stripes $\{X_{\ell - N + 1}^i, \ldots, X_\xi^i\}$ can be recovered from the responses $r_s^i, s \in \mathcal{R}$.
Proof: The code $C * D$ has distance $d_{C*D} = n - (k - t + 1) + 1 \geq |J| + 1$, and it follows that the vector $\sum_{i=0}^{M} E_{t+1} * Y_{i+2} = \sum_{i=0}^{M} X_i * G_{C*E_{t+1}}$ can be recovered for any $\xi \in \mathcal{R}$. By Definition 2, the matrix generating these vectors has rank $Nk$ and thus all $N$ stripes in this window can be recovered.

C. Performance

**Lemma 3.** The PIR rate is given by $R_{PIR}^b \leq \frac{\ell}{\ell + M} \frac{Nk}{Nk - \epsilon n}$, with equality for $k = \frac{Nk}{N - \epsilon n}$.

**Proof:** By definition, $Nk$ information symbols have to be downloaded in each window of $N$ blocks. In each round $d_{C*D} - 1$ symbols of the $k$ desired symbols can be downloaded. The PIR rate is hence given by

$$R_{PIR} = \frac{\ell}{\ell + M} \frac{Nk}{Nk - \epsilon n} \leq \frac{\ell}{(\ell + M) n} \frac{k(n - (k - t - 1))}{Nk - \epsilon n} = \left(1 - \frac{\epsilon}{N}\right) \frac{(\ell - n - (k - t - 1))}{(\ell + M)n}.$$

D. Examples

**Example 1.** Consider the case where $\epsilon = 1$ and $N = 2$. In this case Lemma 2 gives $k = 2k$ and the PIR rate for $\ell \to \infty$ is $R_{PIR} = \frac{1}{2} R_{PIR}^b$ where $R_{PIR}^b$ is the rate achieved by the scheme in [10]. In this case, the same result can be achieved with a trivial scheme that downloads each block twice.

**Example 2.** Let $m = 3$, $M = 1$, $n = 6$, $k = 2$, $t = 1$, $N = 3$ and $\epsilon = 1$. Let $D \in \mathbb{F}_6^{6 \times 6}$ be a random matrix with 6 i.i.d. random codewords from an RS$(n, t)$ code as rows and $J = \{4, 5, 6\}$. Assume the user wants to retrieve the second file $X_2$. With (8) the query matrix is given by

$$D + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_2^0 & \alpha_2^0 & \alpha_2^0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{F}_6^{6 \times 6}.$$

The query $q_j^2$ for the $j$-th server is given by the $j$-th column. In the first iteration the user obtains $X_2^2$. Now assume the second block is lost. In the third and fourth iteration the nodes return $r_{3, j}^2 = \langle q_j^2, [Y_{3,j}, Y_{2,j}]^T \rangle$ and $r_{4, j}^2 = \langle q_j^2, [Y_{4,j}, Y_{3,j}]^T \rangle$.

The user receives

$$r_3^1 = \sum_{s=1}^{m} (D_s * Y_s^2 + D_{M+s} * Y_{2}^2) + [0, 0, 0, Y_{4,4}^2, Y_{4,5}^2, Y_{4,6}^2, Y_{2,4}^2, Y_{2,5}^2, Y_{2,6}^2]$$

$$r_4^1 = \sum_{s=1}^{m} (D_s * Y_{4,4}^2 + D_{M+s} * Y_{2,4}^2) + [0, 0, 0, Y_{4,4}^2, Y_{4,5}^2, Y_{4,6}^2, Y_{2,4}^2, Y_{2,5}^2, Y_{2,6}^2]$$

The distance of $C*D$ is $d_{C*D} = 4$ and treating positions 4 - 6 as erasures gives

$$[Y_{3,4}^2, Y_{3,5}^2, Y_{3,6}^2, Y_{4,4}^2, Y_{4,5}^2, Y_{4,6}^2, Y_{2,4}^2, Y_{2,5}^2, Y_{2,6}^2]$$

$$[G_{C*E_{2,1}}^6 G_{C*E_{2,2}}^6 G_{C*E_{2,3}}^6 \cdots G_{C*E_{2,9}}^6] \cdot \frac{\alpha_2^0 \alpha_2^0 \alpha_2^0 \alpha_2^0 \alpha_2^0 \alpha_2^0 \alpha_2^0 \alpha_2^0 \alpha_2^0}{\alpha_4 \alpha_5 \alpha_6 \alpha_4 \alpha_5 \alpha_6 \alpha_4 \alpha_5 \alpha_6},$$

where $\alpha_2^0 = [\alpha_2^0, \alpha_2^0, \alpha_2^0]$. If this matrix has full rank, i.e., $J$ has the recovering property, the files $X_2^2, X_3^2$ and $X_4^2$ can be recovered. Whether $J$ has the recovering property depends on the choice of evaluation points. The proof that such evaluation points exist can be found in the extended version [17]. By (11) the PIR rate for $\ell \to \infty$ is given by $R_{PIR} = \frac{2}{3}$.

V. PIR with Byzantine Servers and Convolutional Codes

In this section, we consider incorrectly received responses, due to either Byzantine servers or errors during transmission. We focus on constructions that result in a convolutional code of memory $M = 1$, i.e., UM codes. For these codes, the decoder introduced in [16] can efficiently decode up to half the designed extended row distance, by a combination of bounded minimum distance (BMD) decoding in the blocks and trellis based decoding with the Viterbi algorithm. A key step in this algorithm is decoding blocks in the cosets given by successfully decoded neighboring blocks. It is therefore imperative for a good performance to design the code such that these cosets have good distance properties. In the following we describe a scheme that achieves this goal in the PIR setting.

A. Query

We query for two stripes in each block (i.e., $M = 1$) and design the queries such that when one block can be decoded and both stripes are recovered, the neighboring blocks have good distance properties in the corresponding cosets.

Let $D \in \mathbb{F}_2^{2m \times n}$ be as in (4) and $D$ be an RS$(n, t)$ code. The query for the $j$-th server is given by

$$q_j^1 = D_{j, i} + e_1 E_1 + e_{m+1} E_2,$$

where $E_1 = [a_j^{-k}], E_2 = [a_j^{k-t-1}]$ and $e_i$ is the $i$-th standard basis vector.

B. Response

The response to one query consists of $\ell + 1$ parts. In iteration $\xi$ the response of server $j$ is given by

$$r_{\xi, j}^1 = \langle q_j^1, [Y_{\xi,j}, Y_{\xi-1,j}]^T \rangle,$$

where $Y_0 = Y_{\ell+1} = 0$ and $Y_2 = X_{\xi}G$ denotes the matrix storing the $\xi$-th part of every file.

C. Decoding

The user receives

$$r_{\xi, j}^1 \in C*D \quad \forall j \in C*E_1 \quad \forall \xi \in C*E_2$$

$$r_{\xi, j}^1 \in C*E_1 \quad \forall j \in C*E_2$$

$$w_\xi$$,
where \( w_\xi \) denotes the error vector of iteration \( \xi \).

**Lemma 4.** The codes \( C \ast (D + E_1 + E_2) \), \( C \ast (D + E_1) \), and \( C \ast (D + E_2) \) have respective distances \( d_\ast (D + E_1 + E_2) = n - 3k - t + 2 \) and \( d_\ast (D + E_1) = d_\ast (D + E_2) = n - 2k - t + 2 \). The codes \( C \ast D \), \( C \ast E_1 \) and \( C \ast E_2 \) intersect trivially.

**Proof:** An \( RS(n, k, 1) \) code is the evaluation of all polynomials \( f(z) \) with \( \deg(f(z)) \leq k - 1 \) at the evaluation points \( \alpha_j \). Multiplying any polynomials corresponding to the codes \( C \ast D \), \( C \ast E_1 \) and \( C \ast E_2 \) gives

\[
 f(z) \cdot f_D(z) + u_{k+1} z^{-k} + u_{k+1} z^{-k+1} = \sum_{i=0}^{k+t-2} u_i z^i + \sum_{i=0}^{k+t-2} u_i z^i + \sum_{i=0}^{k+t-2} u_i z^i + \sum_{i=0}^{k+t-2} u_i z^i,
\]

where \( u_i \in \mathbb{F} \). Evaluating this polynomial at \( \alpha_j, j \in [n] \), gives a codeword of \( C \ast (D + E_1 + E_2) = RS(n, 3k + t - 1, [\alpha_j^k]) \). By the same argument, it holds that \( C \ast (D + E_1) = RS(n, 2k + t - 1, [\alpha_j^k]) \) and \( C \ast (D + E_2) = RS(n, 2k + t - 1, 1) \). The distances follow from the Singleton bound and the trivial intersection from the distinct powers in the polynomials.

The large number of states makes trellis decoding of the convolutional code infeasible. In [16] an algorithm combining BMD decoding in the blocks and Viterbi decoding on a reduced trellis is given, with decoding complexity only cubic in \( n \), if the complexity of the block decoders is quadratic in \( n \). We give a brief description of this algorithm and show how it can be applied to decode the responses.

1) Decode each received block in \( C_\alpha = C \ast (D + E_1 + E_2) \), an \( RS(n, 3k + t - 1) \) code of distance \( d_\alpha = n - 3k - t + 2 \).

2) From the blocks successfully decoded in step 1) decode \( l_F \) steps forward and \( l_B \) backward (see [16]) in the respective coset \( C \ast (D + E_1) \) or \( C \ast (D + E_2) \). By Lemma 4 these are \( RS(n, 2k + t - 1, [\alpha_j^k]) \) codes and can therefore be decoded up to half their minimum distance \( d_1 = d_2 = n - 2k - t + 2 \).

3) Build reduced trellis and find the maximum-likelihood path with the Viterbi algorithm.

4) By Lemma 4 the codes \( C \ast D \), \( C \ast E_1 \) and \( C \ast E_2 \) intersect trivially, and it follows that the parts of the file \( X^t \) can be recovered uniquely from the codeword corresponding to the most likely path.

**Theorem 2.** If (3) holds, where \( d^\ast_k \) is given by (2) with \( d_\alpha = n - 3k - t + 2 \) and \( d_\beta = n - 2k + t + 2 \), decoding of the responses is successful and the file \( X^t \) is decoded correctly.

**Proof:** By [16] the maximum likelihood path will be in the reduced trellis if (3) holds, which depends on the distance \( d_\alpha \) in each block and the distances \( d_1 \) and \( d_2 \) in the corresponding cosets of the neighboring blocks. For the code given by the responses \( \{r_1, \ldots, r_{l+t+1}\} \) these are shown in Lemma 4. If the path is contained in the trellis, the Viterbi decoder will find it, as it is an ML decoder.

**Corollary 1.** The PIR rate of the scheme is \( R_{PIR} = \frac{\ell k}{(t+1)n} \), with \( n > 3k + t - 1 \) and has error correction capability similar to an optimal \( (n - (k + t - 1), k) \) UM-code.

This result is similar to the scheme of [15] which also has error/erasure correction capability similar to an optimal (MDS) block code of shorter length. This allows for a direct comparison, i.e., in any non-private setting where a block convolutional code performs better than a comparable block code, our scheme will perform better when the privacy requirement is introduced.

**Acknowledgment**

The authors would like to acknowledge Dr. Oliver W. Gnilke for fruitful discussions on this topic and for helpful comments regarding the manuscript.

**References**


