Cryptographic schemes, key exchange, public key

Ted Hurley*

Abstract

General cryptographic schemes are presented where keys can be one-time or ephemeral. Processes for key exchange are derived. Public key cryptographic schemes based on the new systems are established. Authentication and signature schemes are easy to implement. The schemes may be integrated with error-correcting coding schemes so that encryption/coding and decryption/decoding may be done simultaneously.

1 Introduction

This paper introduces cryptographic systems based on operations with randomly chosen vectors, matrices and group ring elements. Keys used may be one-time session keys or ephemeral; as they are easily constructed they may be changed if necessary for each transaction or series of transactions. Key exchange methods are derived. Public key cryptographic schemes based on the new systems are introduced. It is straightforward to include authentication, signature and 'person-in-middle' interference prevention methods based on the schemes. Encryption may be incorporated with error-correcting codes so that encryption/coding and decryption/decoding can be done simultaneously. A public-key scheme can be altered and made private to an individual.

Large pools from which to randomly draw the keys are available; using for example systems of size 101 over Z_p there are of the order of p^{100} different elements from which to choose in the construction of a key.

Features

- Encryption and decryption keys are easy to construct and can be chosen for a one-time session or series of transactions.
- Key exchange schemes are derived.
- Public key cryptographic schemes are developed. These can be altered for private communication and messages then authenticated.
- Authentication, signature and 'person-in-middle' interference prevention methods are provided.
- Encryption and error-correction coding may be integrated into one system. Coding and encryption can complement one another.

When a system is used for one-time session then three transmissions are necessary. A key exchange also requires three transmissions but once a key has been exchanged each transaction naturally then requires just the one transmission.

Layout

The layout of the paper is as follows:

*National Universiy of Ireland Galway, email: Ted.Hurley@NuiGalway.ie

1. Details on various theory required for the constructions are given in section 8 and may be consulted as required. Here also various systems and schemes within which the constructions may be realised are outlined.
2. General encryption methods are introduced in section 2.
3. Key exchange methods are laid out in section 3.
4. Public key encryption methods are given in section 4.
5. Methods to include error-correcting with the cryptography is presented in section 5.
6. Multiple design methods are presented in section 6.
7. Section 7 discusses authentication, signature and 'person-in-middle' solutions.

Basic references for cryptography include [3], [4], [9]. The first two in particular contain much of the algebra required and further basic algebraic material may be obtained in [10].

2 Encrypt message

2.1 General Schemes

Here $R_{n\times n}$ denotes the ring of $n \times n$ matrices, and R^n is the ring of vectors of length n, over a ring R. RG denotes the group ring of the group G over the ring R; for details on general properties of group rings see [10]. The R is usually a field and is often then denoted by F. For details on *group ring matrices* see for example [5]; the set-up and main properties of these are given in section 8.3. They may also be referred to as *RG-matrices* when the group ring in question is specified and are obtained from the embeddings of a group rings into rings of matrices; they include such matrices as circulant matrices, circulant of circulant matrices and similar such.

An RG-matrix, which is of size $|G|\times|G|$, is determined by its first row and is a matrix corresponding to a group ring element, relative to a listing of the elements of G. Two RG-matrices obtained from the same group ring RG (relative to the same listing) are said to be of the same *type*. The RG-matrices commute if and only if G is commutative. Methods to randomly choose singular and non-singular matrices with certain properties from a huge pool of such matrices are given in section 8.5.

Let $\underline{x}$ be a row vector with entries from R. Then the *completion* of $\underline{x}$ in RG (relevant to a particular listing) is the RG-matrix with first row $\underline{x}$. The *rank of a vector*, relative to its completion in a specified group ring, is defined as the rank of its completion; this gives meaning to *kernel of a vector* relative to its group ring completion.

The completion of the vector $\underline{x}$ is denoted by the corresponding capital letter X (without underlining). For $a \in RG$ its image in $R_{n\times n}$ under the embedding of RG into $R_{n\times n}$ is denoted by the corresponding capital letter A. When $\underline{x}$ is a vector to be considered as an element of RG then also use X (without underlining) to denote its image under this embedding.

The following Lemma is immediate.

Lemma 2.1 *Suppose P, Q are RG-matrices with the same first row. Then $P = Q$.*

Thus if $\underline{x}$ is a vector in R^n and A is an RG-matrix of size $n \times n$ then from Lemma 2.1 the completion of $\underline{x}A$ in RG is XA where X is the completion of $\underline{x}$ in RG.

Let $\underline{x}$ be the data to be transmitted secretly from A(lice) to B(ob). The data $\underline{x}$ is arranged so that X is singular with large kernel where X is the completion of $\underline{x}$ in same type of RG-matrix as A (the matrix chosen by A below in 1.); details on how this can be arranged are given below in section 8.5. When X is singular with large kernel then also CX, XC are singular with large kernel for any matrix C.

2.1.1 General set-up

1. A chooses A, a non-singular group ring matrix, and transmits $\underline{x}A$.

2. B chooses B non-singular and transmits BXA.

3. A transmits BX.

4. B works out $B^{-1}BX = X$.

B need not in general be a group ring matrix and even if so it need not be of the same type as A. If B is of the same type as A and X then only the first row of the matrices in 2. and 3. need be transmitted. In 4. only the first row of $B^{-1}BX$ need be calculated as the first row of X give $\underline{x}$. The inverses of A, B should be easily obtainable; pools of matrices from which such matrices may be drawn are given in section 8. When using matrices with certain structures such as RG-matrices the matrix multiplications and vector-matrix multiplications may be performed by convolutional methods.

The matrices A, B here are as large as the vector of data $\underline{x}$ and chosen randomly. The data may also be broken up and multiple vector design schemes implemented as shown in section 6.

Simplified schemes with commuting matrices are derived in the next section 2.1.2; these do not necessarily need RG-matrices.

2.1.2 Commuting schemes

Suppose the large pool of matrices available commute with one another. In these cases simplified schemes may be designed as follows.

Let $\underline{x}$ be data to be transmitted secretly from A(lice) to B(ob).

1. A chooses the matrix A non-singular and $\underline{x}A$ is transmitted.

2. B chooses the matrix B non-singular and transmits $\underline{x}AB$.

3. A transmits $\underline{x}ABA^{-1} = \underline{x}B$.

4. B applies B^{-1} to $\underline{x}B$ to get $\underline{x}$.

Even when the matrices commute, the general scheme of 2.1.1 may still be used.

Schemes with commuting matrices may be achieved using group ring matrices derived from an abelian group ring as for example circulant matrices or circulant of circulant matrices. Section 8.4 discusses types of such matrices which may be used. When such group ring matrices are used the data is arranged so that the group ring matrix formed using $\underline{x}$ as first row is singular with large kernel; how to arrange the data in such a way is discussed later. The non-singular matrices are chosen so that the inverses are immediate or straightforward to calculate.

When some matrices commute the data $\underline{x}$ may also be 'protected' at each end as follows:

1. A chooses A_1, A_2 non-singular and transmits A_1XA_2.

2. B chooses B_1, B_2 non-singular and transmits $B_1A_1XA_2B_2$. It is necessary that $A_iB_i = B_iA_i$ for $i = 1, 2$ but otherwise there are no commuting conditions.

3. A transmits B_1XB_2.

4. B works out X.

If all matrices commute, including X, then the system is the same as above, with A replaced by A_1A_2 and B replaced by B_1B_2.

3 Key exchange

A modification of the general scheme is now set up so that a process may be initiated whereby two intended correspondents can exchange a secret encoder/decoder.

Let $\{\underline{x}, \underline{y}\}$ be vectors so that $\{X, Y\}$ are singular with large kernel and some combination of $\{X, Y\}$ or some combination of $\{X, Y\}$ with a known element or elements is non-singular. Methods to randomly choose such vectors $\{\underline{x}, \underline{y}\}$ are developed in section 8 below.

1. A chooses A non-singular and $\underline{x}$ with large kernel and transmits $\underline{x}A$.
2. B chooses B non-singular and transmits BXA.
3. A transmits BX. B now knows X. A can now repeat the process to get Y secretly to B. Or else:
   (a) B chooses Y with large kernel and B_1 non-singular so that a combination of $\{X, Y\}$ with a known element or known elements is non-singular and transmits B_1Y.
   (b) A transmits B_1YA.
   (c) B transmits YA.
4. Both A and B now have X, Y from which to form the encoding matrix for use between A and B.

When $\underline{x}, \underline{y}$ are known, an *RG*-matrix may be formed from these using a different *RG* from that used for the key exchange. Convolutional methods where appropriate as with group ring matrices may be used for matrix and matrix-vector multiplications. It is sometimes the case that it is sufficient to simply add on a known element or known elements to $\underline{x} + \underline{y}$ to obtain an element whose completion is non-singular. For example $X + Y + 1$ may be known to be non-singular in some systems, see section 8. Knowing the added element(s) gives no information as $\underline{x}, \underline{y}$ are known only to A,B. In section 8.6 it is shown how to randomly choose $\{X, Y\}$ each with large kernel so that a (linear) combination of $\{X, Y\}$ is non-singular and its inverse is easily constructed.

In cases X, Y may be chosen so that small powers of X, Y are zero. This ensures that $\ker X, \ker Y$ are large, see Corollary 8.1 below, and then $\ker XC, \ker CY, \ker CX, \ker YC$ are also large for any matrix C.

When key has been exchanged between A and B, messages between them may then be encrypted directly. When a key has been exchanged it is not necessary to arrange data to be transferred to have large kernel. Messages may also then be encrypted and encoded together as shown later.

The data $\underline{x}, \underline{y}$ may be protected on both sides when some matrices commute; in 1. above, A chooses A_1, A_2 non-singular and $\underline{x}$ with large kernel and transmits A_1XA_2 to which B chooses B_1, B_2 where $A_iB_i = B_iA_i$ and transmits $B_1A_1XA_2B_2$ and process continues as above.

Key exchange with commuting matrices

When the matrices commute the schemes may be simplified as follows.

1. A chooses $\underline{x}$, with large kernel, and A, non-singular, and transmits $\underline{x}A$.
2. B chooses B_1 non-singular and transmits $\underline{x}AB_1$.
3. A transmits $\underline{x}B_1$. At this stage B, and A, know $\underline{x}$. A could proceed to transmit $\underline{y}$ secretly or else:
   (a) B chooses $\underline{y}$ with large kernel and B_2 non-singular and transmits $\underline{y}B_2$.
   (b) A chooses A_1 non-singular and transmits $\underline{y}B_2A_1$.
   (c) B transmits $\underline{y}A_1B$.
4. At this stage both A and B know $\underline{x}, \underline{y}$ from which the combination is formed whose completion is non-singular; this is used as key for transmission(s) between A and B.

Variations

Still using the main ideas, it is clear that many variations on the above schemes can be developed. For example the $\underline{x}$ as X can be 'protected' on both sides by transmitting A_1XA_2 at 1. above and $B_1A_1XA_2B_2$ at 2. where $A_iB_i = B_iA_i$ for $i = 1, 2$; similarly the $\underline{y}$ can be 'protected' on both sides. Methods using series of vectors $\{\underline{x_1}, \underline{x_2}, \dots, \underline{x_r}\}$ and $\{\underline{y_1}, \underline{y_2}, \dots, \underline{y_r}\}$ are presented in section 6.2.

4 Public key

Public key cryptographic methods may be designed by choosing vectors with large kernels from a large pool such that a linear combination of these is non-singular. The participant A constructs a public key as follows.

1. A chooses vectors $\{\underline{x}, \underline{y}\}$ such that their completions $\{X, Y\}$ have large kernels and such that a linear combination of $\{X, Y\}$ is non-singular.
2. A chooses non-singular matrices $\{A_1, A_2\}$ and works out $\{XA_1, YA_2\}$.
3. A has public key (XA_1, YA_2) and private key (X, Y, A_1, A_2).

Suppose now B wishes to communicate $\underline{z}$ to A.

1. B transmits $(\underline{z}XA_1, \underline{z}YA_2)$.
2. A works out $(\underline{z}X, \underline{z}Y)$ and uses the combination $f(X, Y)$ of X, Y to work out $\underline{z}f(X, Y)$ where $f(X, Y)$ is non-singular; from this $\underline{z}$ may be worked out by A.

Methods to randomly choose such $\{\underline{x}, \underline{y}\}$ are shown in section 8.6 and methods to randomly choose such $\{A_1, A_2\}$ appear in various parts of section 8.

The $\underline{x}, \underline{y}$ may be 'protected' on both sides as follows: Step 2. is replaced by: A chooses $\{A_1, A_2, A_3, A_4\}$ and works out (A_1XA_2, A_3YA_4); A then has public key (A_1XA_2, A_3YA_4) and private key $(X, Y, A_1, A_2, A_3, A_4)$. Choosing A_1 from a set of commuting RG-matrices and completing the data $\underline{z}$ to Z relative to RG enables ZX to be recovered by A from ZA_1XA_2 and similarly ZY may be recovered by choosing A_3 from a set of commuting RG_1-matrices where it's not necessary that $G = G_1$.

Details on orthogonal sets of idempotents are given in section 8.6. Here we outline a method of public key construction using full complete orthogonal sets of idempotents. Let $\{E_0, E_1, \dots, E_{n-1}\}$ be an complete orthogonal set of idempotents in $F_{n\times n}$. Thus here each E_i has rank 1 (but this is not necessary in general, see section 8.6).

1. A chooses $J \subset I$ with $|J|$ approximately half of $|I| = n$ and constructs $X = \sum_{j\in J} \alpha_j E_j, Y = \sum_{j\in(I-J)} \beta_j E_j$ with $\alpha_j \neq 0, \beta_j \neq 0$. (Here $\operatorname{rank} X = |J|, \operatorname{rank} Y = |I - J|$. It is enough to choose J so that both X, Y have large kernel.)
2. A chooses $\{A_1, A_2\}$ non-singular and calculates $\{XA_1, YA_2\}$.
3. A has public key (XA_1, YA_2) and private key (X, Y, A_1, A_2).

When B wishes to communicate $\underline{z}$ to A, the process is as follows.

1. B transmits $(\underline{z}XA_1, \underline{z}YA_2)$.
2. A works out $(\underline{z}X, \underline{z}Y)$ and then $\underline{z}(X + Y)$. Now $X + Y$ is invertible and its inverse is easy to calculate, by Lemma 8.5, and A works out $\underline{z}$.

For each n there are many different complete orthogonal sets of idempotents in $F_{n\times n}$. It is not necessary that the particular complete orthogonal set used by A in constructing her public key be known to the world so in fact an additional step before step 1. could be:

0. A chooses a complete orthogonal set of idempotents $\{E_0, E_1, \dots, E_{n-1}\}$.

Convolutional methods where appropriate may be used for matrix and vector-by-matrix multiplications. The public keys may be changed from time to time. Errors $(zXA_1 + \alpha, \underline{z}YA_2 + \beta)$ with $\alpha \neq 0$ or $\beta \neq 0$ in transmitting $(\underline{z}XA_1, \underline{z}YA_2)$ are easily detected unless $\alpha = \gamma XA_1$ and $\beta = \gamma YA_2$ which is extremely unlikely. This does not prevent an intruder from trying to falsify a message but a method to prevent this is given in section 4.1 below.

4.1 From public to private

Suppose now A has public key $(\underline{x}A_1, \underline{y}A_2)$. This can be made into a private key for B with which messages from B only to A may be received:

- B chooses $\{B_1, B_2\}$ non-singular and transmits (B_1XA_1, B_2YA_2).
- A chooses $\{A_{B_1}, A_{B_2}\}$ and transmits $(B_1XA_{B_1}, B_2YA_{B_2})$.
- B has key (XA_{B_1}, YA_{B_2}) with which to send messages to A.

Some simplification is possible when matrices commute.

- B chooses $\{B_1, B_2\}$ non-singular and transmits $(\underline{x}A_1B_1, \underline{y}A_2B_2)$.
- A chooses $\{A_{B_1}, A_{B_2}\}$ and transmits $(\underline{x}A_{B_1}, \underline{y}A_{B_2})$.
- B has (private) key $(\underline{x}A_{B_1}, \underline{y}A_{B_2})$ with which to send messages to A.

Suppose now B has key $(\underline{x}A_{B_1}, \underline{y}A_{B_2})$ with which to send message to A. Using this key, B sends message $\underline{z}$ to A. Then A can work out $\underline{z}XA_{B_1}$ and check that message has not been interfered with; an intruder would need to know XA_{B_1} in order to change message that would not be discovered in a check.

4.2 Partial public key

It is useful at times, in particular for authentication and signature schemes, for a participant to make public a 'key' of the form $\underline{y}B$ where $\underline{y}$ has large kernel and B is invertible. Now $\underline{y}B$ cannot be inverted and so may not be used as a key itself. It could be used for a message authentication scheme or signature scheme.

This can be made private to another particular user by methods similar to those used in section4.1. Suppose B has published $\underline{y}B$ where $\underline{y}$ has large kernel and B is invertible where $\{\underline{y}, B\}$ are kept private.

- A chooses A and transmits AYB.
- B chooses B_A and transmits AYB_A.
- A uses $\underline{y}B_A$ with B.

A simplification using commuting matrices may be initiated similar to section 4.1.

4.3 Multiple design for public key

In the above scheme, vectors $\{\underline{x}, \underline{y}\}$ such that their completions $\{X, Y\}$ have large kernels and such that a linear combination of $\{X, Y\}$ is non-singular are chosen. More generally vectors $\{\underline{x_1}, \underline{x_2}, \ldots, \underline{x_r}\}$ such that their completions $\{X_1, X_2, \ldots, X_r\}$ have large kernels and such that a linear combination of $\{X_1, X_2, \ldots, X_r\}$ is non-singular may be chosen. However this increases the amount of data to be transmitted as each $\underline{z}X_iA_i$ needs to be transmitted. However again one of these could be laid aside authentication; for example a triple of form $(\underline{x}A_1, \underline{y}A_2.\underline{p}A_3)$ each with large kernel such that a linear combination of $\{X, Y, P\}$ is non-singular is used but $\underline{x}A_1 = \underline{x}A_B$ is private for B only to be used as a check; when the message $\underline{z}$ is worked out, $\underline{z}XA_B$ is used as a message authentication check. An original $\underline{x}A_1$ may be altered to $\underline{x}A_B$ by methods similar to those in section 4.1.

5 Cryptography + error-correction

The cryptographic systems may be used simultaneously with error-correcting systems. A basic general reference for coding theory is [2].

Let $\underline{x_1}$ be $1 \times r$ data to be transmitted securely (with encryption) and safely (with error coding) from A to B. Let G be a generator $r \times n$ matrix of an error-correcting code and $\underline{x} = \underline{x_1}G$. When matrices don't necessarily commute proceed as follows:

1. A works out $\underline{x} = \underline{x_1}G$ chooses A non-singular and transmits $\underline{x}A$.
2. B chooses B non-singular and transmits BXA.
3. A transmits BX.
4. B calculates $B^{-1}BX = X$ to get $\underline{x}$ which may have errors in transmission. B decodes the obtained $\underline{x}$ to get $\underline{x_1}$.

If using RG-matrices of the same type only the first row of matrices need be worked out.

In general it is shown in Proposition 8.2 that if G is the generator matrix of an (n, r) code which has rank r and G is a zero-divisor code (as are cyclic and similar codes, see [8]) then the completion of $\underline{x} = \underline{x_1}G$ has rank at most r and so $\dim \ker$ of the completion of $\underline{x}$ is $\geq (n - r)$.

When encryption/decryption matrices to be chosen commute the following simplified method may be used:

1. A works out $\underline{x} = \underline{x_1}G$, chooses A non-singular and transmits $\underline{x}A$.
2. B chooses B non-singular and transmits $\underline{x}AB$.
3. A transmits $\underline{x}B$.
4. B works out $\underline{x}$ which may have errors in the transmissions and decodes to $\underline{x_1}$.

The code determined by G is an (n, r) code with rank r. When for example G is cyclic then G can be taken as the first r rows of a circulant matrix which has rank r. Then the completion of $\underline{x} = \underline{x_1}G$ is a circulant matrix of rank at most r. The kernel then of this completion is of dimension at least $(n - r)$. See section 8.8 for details on these aspects.

5.1 Key exchange with coding

Modify the methods of section 3 as follows to include error-correcting codes.

Let $\{\underline{x_1}, \underline{y_1}\}$ be $1 \times r$ vectors so that $\{X_1, Y_1\}$ are singular with large kernel and some combination of $\{X_1, Y_1\}$ with a known element or elements is non-singular. Methods for randomly being able to choose such vectors $\{\underline{x_1}, \underline{y_1}\}$ are discussed in section 8 below.

Let G, L be generator $r \times n$ matrices of (n, r) error-correcting codes.

1. A chooses A non-singular and $\underline{x_1}$ and transmits $\underline{x_1}GA$.
2. B chooses B and transmits BXA where X is completion of $\underline{x} = \underline{x_1}G$.
3. A transmits BX.
4. B now knows $\underline{x}$ which may contain errors but is decoded to $\underline{x_1}$.
    (a) B chooses B_1 non-singular and $\underline{y_1}$ so that the completion of a combination of $\{\underline{x_1}, \underline{y_1}\}$ with a known element or known elements is non-singular and transmits B_1Y where $\underline{y} = \underline{y_1}L$.
    (b) A chooses A and transmits B_1YA.
    (c) B transmits YA. A knows $\underline{y}$ with possible errors and decodes to $\underline{y_1}$.
5. Both A and B now have $\underline{x_1}, \underline{y_1}$ from which to form the encoding matrix as in section 3.

6 Multiple vector design

The data to be transmitted is broken as $(\underline{x_1}, \underline{x_2}, \ldots, \underline{x_r})$. The $\underline{x_i}$ need not be of the same length and are arranged so that the X_i are singular except for possibly a relatively very small number of these.

6.1 General schemes

B_i and A_i are group ring matrices and X_i and A_i are of the same type.

1. A chooses $\{A_1, A_2, \ldots, A_r\}$ non-singular and transmits $(\underline{x_1}A_1, \underline{x_2}A_2, \ldots, \underline{x_r}A_r)$
2. B chooses $\{B_1, B_2, \ldots, B_r\}$ non-singular and transmits $(B_1X_1A_1, B_2X_2A_2, \ldots, B_rX_rA_r)$.
3. A transmits $(B_1X_1, B_2X_2, \ldots, B_rX_r)$.
4. B reads $(\underline{x_1}, \underline{x_2}, \ldots, \underline{x_r})$ as the first row of $(X_1, X_2, \ldots, X_r)$.

The matrices do not need to commute and B_i need not be of the same type as A_i, X_i. If B_i is of the same type as X_i, A_i then only the first rows of the matrices need be transmitted in 2. 3. above. In these cases convolution methods for multiplication may be used.

6.1.1 Schemes with some matrices commuting

Here we have matrices A_i, B_i with $A_iB_i = B_iA_i$ for each i; it is not necessary that $A_iB_j = B_jA_i$ for $i \neq j$ nor that $A_iA_j = A_jA_i$ for any i, j.

1. A chooses $\{A_1, A_2, \ldots, A_r\}$ non-singular and transmits $(\underline{x_1}A_1, \underline{x_2}A_2, \ldots, \underline{x_r}A_r)$
2. B chooses $\{B_1, B_2, \ldots, B_r\}$ non-singular and transmits $(\underline{x_1}A_1B_1, \underline{x_2}A_2B_2, \ldots, \underline{x_r}A_rB_r)$.
3. A transmits $(\underline{x_1}B_1, \underline{x_2}B_2, \ldots, \underline{x_r}B_r)$.
4. B reads $(\underline{x_1}, \underline{x_2}, \ldots, \underline{x_r})$.

6.2 Key exchange with multiple vectors and matrices

Key exchange with multiple vector choices may be achieved as follows:

Let $\{\underline{x_1}, \underline{x_2}, \ldots, \underline{x_r}\}$ and $\{\underline{y_1}, \underline{y_2}, \ldots, \underline{y_r}\}$ be sets of vectors where for each i, $\underline{x_i}$ has the same length as $\underline{y_i}$; these are chosen randomly so that X_i, Y_i are singular (except possibly for a relatively small number of them) and some combination of X_i, Y_i is non-singular or some combination of X_i, Y_i with a known element or known elements is non-singular.

1. A chooses $\{A_1, A_2, \ldots, A_r\}$ non-singular and $(\underline{x_1}, \underline{x_2}, \ldots, \underline{x_r})$ and transmits $(\underline{x_1}A_1, \underline{x_2}A_2, \ldots, \underline{x_r}A_r)$.
2. B chooses $\{B_1, B_2, \ldots, B_r\}$ non-singular and transmits $(B_1X_1A_1, B_2X_2A_2, \ldots, B_rX_rA_r)$.
3. A transmits $(B_1X_1, B_2X_2, \ldots, B_rX_r)$.
4. B now knows $(X_1, X_2, \ldots, X_r)$. A can now repeat the process to get $(Y_1, Y_2, \ldots, Y_r)$ secretly to B. Or else:
   (a) B chooses $(\underline{y_1}, \underline{y_2}, \ldots, \underline{y_r})$ so that a combination of X_i, Y_i or a combination of X_i, Y_i with a known element or known elements is non-singular.
   (b) B chooses $\{B'_1, B'_2, \ldots, B'_r\}$ non-singular and transmits $(B'_1Y_1, B'_2Y_2, \ldots, B'_rY_r)$.
   (c) A chooses $\{A'_1, A'_2, \ldots, A'_r\}$ non-singular and transmits $(B_1Y_1A'_1, B_2Y_2A'_2, \ldots, B_rY_rA'_r)$.
   (d) B transmits $(Y_1A'_1, Y_2A'_2, \ldots, Y_rA'_r)$.
5. Both A and B now have the X_i, Y_i for each i from which to form the secret encryption matrices.

6.3 Key exchange with multiple vectors and coding

Key exchange with multiple vector choices and coding may be achieved as follows:

Let $\{\underline{\underline{x_1}}, \underline{\underline{x_2}}, \dots, \underline{\underline{x_r}}\}$ and $\{\underline{\underline{y_1}}, \underline{\underline{y_2}}, \dots, \underline{\underline{y_r}}\}$ be sets of vectors where for each i, $\underline{\underline{x_i}}$ has the same length as $\underline{\underline{y_i}}$; these are chosen randomly so that their completions $\underline{X_i}, \underline{Y_i}$ are singular except possibly for a small number of them and some combination of their completions is non-singular or some combination of the completions with known elements are non-singular. Define $\underline{x_i} = \underline{\underline{x_i}} G_i, \underline{y_i} = \underline{\underline{y_i}} K_i$ for appropriately sized generator matrices G_i, K_i of error-correcting codes.

1. A chooses $\{A_1, A_2, \dots, A_r\}$ non-singular and transmits $(\underline{x_1}A_1, \underline{x_2}A_2, \dots, \underline{x_r}A_r)$.
2. B chooses $\{B_1, B_2, \dots, B_r\}$ non-singular and transmits $(B_1X_1A_1, B_2X_2A_2, \dots, B_rX_rA_r)$.
3. A transmits $(B_1X_1, B_2X_2, \dots, B_rX_r)$.
4. B now knows $(\underline{x_1}, \underline{x_2}, \dots, \underline{x_r})$ with possible errors and decodes this to $(\underline{\underline{x_1}}, \underline{\underline{x_2}}, \dots, \underline{\underline{x_r}})$. A can now repeat the process to get $(\underline{\underline{y_1}}, \underline{\underline{y_2}}, \dots, \underline{\underline{y_r}})$ secretly to B. Or else:
   (a) B chooses $(\underline{\underline{y_1}}, \underline{\underline{y_2}}, \dots, \underline{\underline{y_r}})$ so that a combination of $\underline{X_i}, \underline{Y_i}$ or a combination of $\underline{X_i}, \underline{Y_i}$ with a known element or elements is non-singular.
   (b) B chooses $\{B'_1, B'_2, \dots, B'_r\}$ non-singular and transmits $(B'_1\underline{y_1}, B'_2\underline{y_2}, \dots, B'_r\underline{y_r})$.
   (c) A chooses $\{A'_1, A'_2, \dots, A'_r\}$ non-singular and transmits $(B_1Y_1A'_1, B_2Y_2A'_2, \dots, B_rY_rA'_r)$.
   (d) B transmits $(Y_1A'_1, Y_2A'_2, \dots, Y_rA'_r)$. A then knows $(\underline{y_1}, \underline{y_2}, \dots, \underline{y_r})$ with possible errors and decodes to $\underline{\underline{y_1}}, \underline{\underline{y_2}}, \dots, \underline{\underline{y_r}})$.
5. Both A and B now have the $\underline{\underline{x_i}}, \underline{\underline{y_i}}$ for each i from which to form the secret encryption matrices.

7 Who is there?

Authentication and/or signature methods may be set up in the usual way when key exchange and/or public key schemes have been established. Section 4.1 shows how public key may in a unique way be used to establish that message is actually emanating from a correspondent A; the constituents of the public key for B are changed so the new key may be used only by a particular A.

Without using public key or key exchange one or both of the following may be requirements.

- In a message exchange from A to B it may be the case that a response from B is required. In certain situations then A requires to know that no one else is responding pretending to be B. ('Person-in-middle' problem.)
- B requires to know that message purporting to come from A is actually from A.

To prevent these 'person-in-middle' problems proceed as follows. Each person X must have a 'key' which is of the form $\underline{y_X}X$ where $\underline{y_X}$ has large kernel. This must be known to and trusted by the person with whom the contact is to be made but may be public. This is a 'partial' public key as discussed in section 4.2.

7.1 Prevent Eve pretending to be B

E(ve), an eavesdropper, looking in at the communications in section 2.1.1 or 2.1.2 can see $\underline{x}A$ and pretends to be B. (S)he then applies E to get EXA in 2.1.1 or $\underline{x}AE$ in 2.1.2, which is then transmitted to A who applies A^{-1} and gives back EX or $\underline{x}E$ to E who can then read off $\underline{x}$.

A wants to communicate $\underline{x}$ secretly to B. As stated the key for B is constructed from a vector $\underline{y}$ and a non-singular matrix B and only the product $\underline{y}B$ is known to A but it may be public. $\underline{y}$ should be chosen so that its completion Y is singular and has large kernel. However $\underline{y}B$ is not a public key for B in general as it does not have an inverse.

Use the convention that matrices A and A_* for suffices $*$ are matrices chosen and applied by A(lice) and B and B_* are matrices chosen and applied by B(ob).

7.1.1 With Commuting matrices

Suppose the matrices commute.

1. B chooses signature key $\underline{y}B$ which is revealed.
2. A chooses $\{A, A_1\}$ and sends out $(\underline{x}A, \underline{y}BA_1)$.
3. B chooses $\{B_1, B_2\}$ and transmits $(\underline{x}AB_1, \underline{y}A_1B_2)$.
4. A works out $(\underline{x}B_1, \underline{y}B_2)$ and transmits $(\underline{x}B_1 - \underline{y}B_2)$.
5. B works out $(\underline{x} - \underline{y}B_2B_1^{-1})$ and $\underline{y}B_2B_1^{-1}$ and adds the two to get $\underline{x}$.

In fact for 5. $\underline{y}B_2B_1^{-1}$ can be worked out when B_1, B_2 are chosen at 3. . Now A never knows $\underline{y}$ in this set-up so B may use the same $\underline{y}B$ in communicating with another.

At point 4. A knows $\underline{y}B_2$ and may use this it in further transactions from A to B avoiding some transmissions at points 2. , 3. above. In a sense then when A knows $\underline{y}B_2$ it may be as a 'key' for transmissions from A to B and may be used as a non-public signature of B for A only. Some simplification can be initiated when B is not worried that A may find $\underline{y}$.

7.1.2 With matrices which may not commute

Similar schemes using non-commuting matrices are developed as follows.

A is required to transmit $\underline{x}$ to B and make sure that an eavesdropper may not pretend to be B. The matrices A_* and B_* need not be of the same type, that is, need not be formed from the same group ring.

1. B chooses $\underline{y}$ and B and circulates $\underline{y}B$ (keeping $\underline{y}$ and B secret).
2. A chooses A, A_1 and transmits $(\underline{x}A, A_1YB)$.
3. B chooses B_1, B_2 and transmits (B_1XA, A_1YB_2)
4. A works out (B_1X, YB_2) and transmits $(B_1X - YB_2)$.
5. B works out $B_1^{-1}(B_1X - YB_2) = X - B_1^{-1}YB_2$ and adds it to $B_1^{-1}YB_2$, which may be worked out previously, to get X.

At point 4. A knows YB_2 which can be used for further transactions from A to B.

Variations on the above are easily constructed and designed.

7.2 To be sure

Suppose now A communicate with B and B wishes to be sure that the message is from A.

7.2.1 Where from, commuting

Each participant X has $\underline{y_X}X$, where X is invertible. When RG-matrices are used the completion of $\underline{y_X}$ should be singular with large kernel. $\underline{y_X}$ and X are kept secret.

1. A chooses A_1 and transmits $\underline{y_A}A_1$.
2. B chooses B_1 and transmits $\underline{y_A}A_1B_1$.
3. A transmits $\underline{y_A}AB_1$ and B checks this.

(At stage 2. B can work out $\underline{y_A}AB_1$ for checking at 3.)

7.2.2 Where from, non-commuting

Suppose A wishes to communicate with B and B wishes to be sure that the message is from A. Each participant X publishes XY_X, where X is invertible and Y_X is singular with large kernel. Y_X and X are kept secret.

1. A chooses A_1 and transmits A_1Y_A.
2. B chooses B_1 and transmits $A_1Y_AB_1$. (At this stage B can work out AY_AB_1.)
3. A transmits AY_AB_1 and B checks this.

7.3 Combined

The methods of 7.1, 7.2 may be combined as required or necessary. A wishes to communicate with B; A requires that an eavesdropper may not pretend to be B and B requires a signature so that (s)he knows the message is from A. The methods are fairly straightforward and details are omitted.

7.4 Multiple vector design: Prevention

The authentication, signature methods devised above may also be extended to multiple vector design. We outline just one of the methods.

7.4.1 Prevent E pretending

It is required when A communicates with B that E may not reply to A succeeding in pretending to be B.

1. B has a key $(\underline{y_1}B_1, \underline{y_2}B_2, \ldots, \underline{y_s}B_r)$ which is revealed at a particular time and known and trusted by A.
2. A sends out $((\underline{x_1}A_1, \underline{x_2}A_2, \ldots, \underline{x_r}A_r), (\underline{y_1}B_1A'_1, \underline{y_2}B_2A'_2, \ldots, \underline{y_s}B_rA'_r))$ where $(\underline{x_1}, \underline{x_2}, \ldots, \underline{x_r})$ is the data to be transmitted and the size of x_i is the same as that of y_i.
3. B chooses $(B'_1, B'_2, \ldots, B'_r)$ and transmits $((\underline{x_1}A_1B'_1, \underline{x_2}A_2B'_2, \ldots, \underline{x_r}A_rB'_r), (\underline{y_1}A'_1, \underline{y_2}A'_2, \ldots, \underline{y_r}A'_r))$.
4. A transmits $(\underline{x_1}B'_1, \underline{x_2}B'_2, \ldots, \underline{x_r}B'_r) - (\underline{y_1}, \underline{y_2}, \ldots, \underline{y_r})$.
5. B works out $(\underline{x_1}, \underline{x_2}, \ldots, \underline{x_r}) - (\underline{y_1}B_1'^{-1}, \underline{y_2}B_2'^{-1}, \ldots, \underline{y_r}B_r'^{-1})$ and adds this to $(\underline{y_1}B_1'^{-1}, \underline{y_2}B_2'^{-1}, \ldots, \underline{y_r}B_r'^{-1})$, which has already been worked out, to get $(\underline{x_1}, \underline{x_2}, \ldots, \underline{x_r})$.

7.5 Authentication, signature, + coding

Authentication and signature with coding may similarly be implemented. The details are omitted. Basically first of all the data is encoded as $\underline{x} = \underline{x_1}G$. Then when $\underline{x}$ is received with possible errors it is decoded to $\underline{x_1}$.

8 Theory

8.1 Vector by matrix multiplication

Much is contained in the literature on vector-matrix/matrix-vector multiplication. The multiplication can be very fast when the matrix has a structure as for example if the matrix is an RG-matrix; a circulant matrix is such an example. Group ring matrices of the groups C_n, C_2^n, C_p^n and in general abelian groups are particularly suitable. Vector-matrix multiplication in these cases using fast Fourier transform or Walsh-Hadamard fast transform (for FC_p^n) can be done in $0(n \log n)$ time.

The multiplication can be done in $0(n\log n))$ time for FG-matrices when G is a finite supersolvable group; this comes from Baum's Theorem [1] which states that every supersolvable finite group has a DFT (Discrete Fourier Transform) algorithm running in $O(n\log n)$ time. This more general notion is not discussed further here

8.2 Rank and nullity

Knowledge of a singular matrix and a product of this singular matrix by a non-singular matrix does not lead to knowledge of the non-singular matrix. It is desirable that the kernel of the singular matrix be relatively large. The nullity of an $n\times n$ matrix with $A^t=0$ is greater than or equal to $\frac{n}{t}$; see Corollary 8.1 below. For large n and relatively small t a solution of a system of equations as $AX=B$, with X as indeterminates or $\underline{x}A=\underline{b}$ with indeterminates $\underline{x}$ then has many possible solutions. If A has (relatively) small rank then so does AY and YA for any Y as $YA\leq\min\{\operatorname{rank}X,\operatorname{rank}A\}$ and $AY\leq\min\{\operatorname{rank},Y,\operatorname{rank}A\}$.

Lemma 8.1 *Let A be an $n\times n$ matrix such that $A^t=0$. Then* $\operatorname{rank}A\leq\frac{n(t-1)}{t}$.

Proof: Note first that for $n\times n$ matrices $\operatorname{rank}PQ\geq\operatorname{rank}P+\operatorname{rank}Q-n$.

Suppose then $\operatorname{rank}A>\frac{n(t-1)}{t}$. We now show by induction that $\operatorname{rank}A^r>\frac{n(t-r)}{t}$ for $1\leq r\leq t$. The case $r=1$ is part of the hypothesis. Suppose then $\operatorname{rank}A^k>\frac{n(n-k)}{t}$ for $1\leq k<t$. Hence $\operatorname{rank}A^{k+1}=\operatorname{rank}AA^k\geq\operatorname{rank}A+\operatorname{rank}A^k-n>\frac{n(t-1)}{t}+\frac{n(t-k)}{t}-n=\frac{n(t-(k+1))}{t}$ as required.

Now $AA^{t-1}=0$ implies that $A^{t-1}\subseteq\ker A$ and so $\operatorname{rank}A^{t-1}\leq\dim\ker A$. But $\operatorname{rank}A+\dim\ker A=n$ implies $\dim\ker A=n-\operatorname{rank}A=n-\frac{n(t-1)}{t}=\frac{n}{t}$ and so $\operatorname{rank}A^{t-1}\leq\frac{n}{t}$. However letting $r=t-1$ in $\operatorname{rank}A^r>\frac{n(t-r)}{t}$ implies $\operatorname{rank}A^{t-1}>\frac{n}{t}$ which is a contradiction. Hence $\operatorname{rank}A\leq\frac{n(t-1)}{t}$. □

Corollary 8.1 *Suppose $A^t=0$ for an $n\times n$ matrix. Then* $\dim\ker A\geq\frac{n}{t}$.

Proof: This follows from the Lemma since $\operatorname{rank}+\dim\ker=n$. □

The following may also be shown but is not relevant here:

Lemma 8.2 *Suppose A is an $n\times n$ matrix with $A^t=0$. Suppose also* $\operatorname{rank}A=\frac{n(t-1)}{t}$. *(This is largest it can be by Lemma 8.1.). Then* $\operatorname{rank}A^{t-1}=\frac{n}{t}$. *In particular this implies $A^{t-1}\neq 0$.*

8.3 RG-matrices

An RG-matrix is a matrix corresponding to a group ring element in the isomorphism from the group ring into the ring of $R_{n\times n}$ matrices, see for example [5]. Specifically suppose $w=\sum_{i=1}^{n}\alpha_{g_i}g_i\in RG$ where $G=\{g_1,g_2,\ldots,g_n\}$ is a listing of the elements of G. The RG-matrix of w denoted by $M(RG,w)$ is defined as follows:

$$\begin{pmatrix}\alpha_{g_1^{-1}g_1} & \alpha_{g_1^{-1}g_2} & \alpha_{g_1^{-1}g_3} & \cdots & \alpha_{g_1^{-1}g_n}\\ \alpha_{g_2^{-1}g_1} & \alpha_{g_2^{-1}g_2} & \alpha_{g_2^{-1}g_3} & \cdots & \alpha_{g_2^{-1}g_n}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \alpha_{g_n^{-1}g_1} & \alpha_{g_n^{-1}g_2} & \alpha_{g_n^{-1}g_3} & \cdots & \alpha_{g_n^{-1}g_n}\end{pmatrix}$$

The matrix is in $R_{n\times n}$ and depends on the listing of the elements. Changing the listing changes the matrix; if A,B are RG-matrices for the element $w\in RG$ relative to different listings then B may be obtained from A by a sequence of [interchanging two rows and then interchanging the corresponding two columns].

Given the entries of the first row of an RG-matrix, and a listing, the entries of the other rows are determined from the multiplication of the elements of G and each row and each column is a permutation of the first row.

Theorem 8.1 *Given a listing of the elements of a group G of order n there is a bijective ring homomorphism between RG and the $n \times n$ RG-matrices. This bijective ring homomorphism is given by $\sigma : w \mapsto M(RG, w)$.*

An RG-matrix for a cyclic group G is a circulant matrix; an RG-matrix when G is a dihedral group is one of the form $\left(\begin{smallmatrix} A & B \\ B & A \end{smallmatrix}\right)$ (in a natural listing of the elements of G), where A is circulant and B is reverse circulant.

For $w \in RG$ the corresponding capital letter W denotes the image of w in the ring of $R_{n\times n}$ matrices, relative of course to a particular listing of the elements of G. For a vector $\underline{x} \in R^n$ and a fixed listing of a group G by convention the capital letter X, without underlining, denotes the completion of $\underline{x}$. Say $w \in RG$ is singular if and only if $W \in R_{n\times n}$ is a singular matrix and w is non-singular if and only if W is a non-singular matrix. Thus when R is a field w is singular if and only if w is a zero-divisor in RG, and w is non-singular if and only if w is a unit in RG, [5].

8.4 Commuting matrices

Matrices that commute with one another include group ring matrices corresponding to group rings of abelian groups. Convenient such group ring matrices include:

1. Circulant matrices over any field; in particular circulant matrices over finite fields such as Z_p for p a prime.

2. RG-matrices from $RG = Z_2C_2^n$. An element $w = \sum_{i=0}^{2^n-1} \alpha_i a_i$ is invertible if and only if $\sum_{i=0}^{2^n-1} \alpha_i = 1$, that is, if and only if there are an odd number of non-zero coefficients in w. For say $n = 1024$ there are 2^{1023} such invertible elements and 2^{1023} elements whose square is zero.

3. Matrices from $Z_pC_p^n$. Let $w = \sum_{i=0}^{p^n-1} \alpha_i a_i \in \mathbb{Z}_pC_p^n$, with $\alpha_i \in \mathbb{Z}_p, a_i \in C_p^n$. Since $w^p = \sum \alpha_i$, it follows that w is invertible if and only if $\sum_{i=0}^{p^n-1} \alpha_i \neq 0$. If this sum $s = \sum \alpha_i$ is zero then w is a zero-divisor with $w^p = 0$ and if this sum $s \neq 0$ then $w^{-1} = s^{-1}w^{p-1}$.

   For say $n = 102$ there are $p^{101}(p-1)$ such invertible elements and $p^{101} - 1$ such non-zero elements which are zero-divisors satisfying $w^p = 0$. It is easy to choose randomly an invertible element whose inverse is easy to construct or a zero-divisor element with relatively small power equal to zero.

The types of matrices used for the designs and for the transmissions of vectors need not be the same.

8.5 Construction methods

For our constructions it is required to randomly choose, from a large available pool, matrices and vectors of the following types:

- Singular matrices A with large kernel.
- Non-singular matrices A such that the inverse of A is easy to compute;
- Vectors $\underline{x}, \underline{y}$ such that X, Y have large kernels and a combination of X, Y with a known element or known elements is non-singular, the inverse of which is easy to obtain.

Further:

- Given data $\underline{x}$ it is required to construct $\underline{\underline{x}}$ from which $\underline{x}$ may directly be obtained and for which the completion of $\underline{\underline{x}}$ is singular with large kernel.

Here we show how such constructions may be obtained in various group ring matrices.

8.5.1 In $\mathbb{Z}_2 C_2^n$

Consider $\mathbb{Z}_2 C_2^n$. An element $w = \sum_{i=0}^{2^n-1} \alpha_i a_i \in \mathbb{Z}_2 C_2^n$ satisfies $w^2 = \sum_{i=0}^{2^n-1} \alpha_i$ and so $w^2 = 0$ or $w^2 = 1$ according to whether the sum of the coefficients of w is even or odd. When the sum is even then $w^2 = 0$ and so indeed w is singular with large kernel by Corollary 8.1. In $\mathbb{Z}_2 C_2^n$ it is easy to arrange for any data $\underline{x}$ that if $\underline{x}^2 \neq 0$ then adding one known element to $\underline{x}$ ensures the square of the data is zero. When $\underline{x}^2 = 0$ then $\operatorname{rank} X$, where x is the completion of $\underline{x}$ is at most $\frac{n}{2}$ and thus $\dim \ker X \geq \frac{n}{2}$; for large n this ensures $\dim \ker X$ is large. Thus there are at least $2^{\frac{n}{2}}$ solutions in $\underline{z}$ to $X\underline{z} = \underline{b}^{\mathrm{T}}$ or $XZ = P$ for unknown matrix Z.

Thus in $\mathbb{Z}_2 C_2^n$:

- Random Matrices X may be chosen such that $X^2 = 0$ and so has large kernel;
- Random Matrices A may be chosen such that $A^2 = 1$ and so the inverse is easy to obtain.
- Random $\underline{x}, \underline{y}$ may be chosen so that $X^2 = 0, Y^2 = 0$ and then both X, Y are singular with large kernel. Combinations such as $X + Y + 1$, $X + Y + H$ where $h \in C_2^n$ and $X + Y + w$ where w has an odd number of non-zero terms have their squares equal to 1.
- If $\underline{x}$ is any vector considered in $\mathbb{Z} C_2^n$ then either $X^2 = 0$ and has large kernel or else adding an element h of C_2^n (h could be the identity) ensures $(X + H)^2 = 0$, or more generally adding an element w with an odd number of non-zero terms ensures $(X + W)^2 = 1$.

For any ring R, an RC_2 matrix is one of the form $\begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix}$ with $\alpha, \beta \in R$. An RC_2^n matrix for $n \geq 2$ is one of the form $\begin{pmatrix} A_{n-1} & B_{n-1} \\ B_{n-1} & A_{n-1} \end{pmatrix}$ where A_{n-1}, B_{n-1} are RC_2^{n-1}-matrices. An RC_2^n-matrix is completely determined by its first row as is any RG-matrix.

Any RC_2^n-matrix is diagonalised by the Walsh-Hadamard $2^n \times 2^n$ matrix which is defined as follows. The Walsh-Hadamard 2×2 matrix is $W_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ and for $n \geq 2$ the Walsh-Hadamard $2^n \times 2^n$ matrix is $W_{2^n} = \begin{pmatrix} W_{2^{n-1}} & W_{2^{n-1}} \\ W_{2^{n-1}} & -W_{2^{n-1}} \end{pmatrix} = W_2 \otimes W_{2^{n-1}}$ where $\otimes$ denotes tensor product. It is known that the Walsh-Hadamard transformation can be performed in time $O(m \log m)$ ($m = 2^n$) and thus vector and matrix operations with RC_2^n-matrices can be done in $O(m \log m)$ time. Thus using $\mathbb{Z}_2 C_2^n$ the constructions may be done in $O(m \log m)$ time using Walsh-Hadamard transformations.

8.5.2 In $\mathbb{Z}_p C_p^n$

Consider now the data $\underline{x} = (\alpha_0, \alpha_1, \ldots, \alpha_{p^n-1})$ to be in $\mathbb{Z}_p C_p^n$, that is $\underline{x} = \sum_{i=0}^{p^n-1} \alpha_i g_i$ where $\{g_0, g_1, \ldots, g_{p^n-1}\}$ are the elements of $\mathbb{Z}_p^n$ and $\alpha_i \in Z_p$. Each g_i satisfies $g_i^p = 1$. Then $\underline{x}^p = \sum_{i=0}^{p^n-1} \alpha_i^p = \sum_{i=0}^{p^n-1} \alpha_i = \epsilon(\underline{x})$ where $\epsilon(\underline{x})$ denotes the augmentation of $\underline{x}$. If $\epsilon(\underline{x}) = 0$ then $\underline{x}^p = 0$. If $\epsilon(\underline{x}) \neq 0$ then $(\underline{x} - \epsilon(\underline{x})g)^p = 0$ for any $g \in C_p^n$. More generally $(\underline{x} + \sum_{j \in J} b_j g_j)^p = 0$ when $\sum_{j \in J} b_j = -\epsilon(\underline{x})$ for $J \subset \{0, 1, \ldots, p^n - 1\}$.

Thus it is easily arranged for the data $\underline{x}$ to satisfy $\underline{x}^p = 0$ by adding a known element or known elements as necessary. If now $\underline{x}^p = 0$ then the completion X of $\underline{x}$ has $\dim \ker X \geq \frac{n}{p}$. Hence any system of equations $X\underline{z} = \underline{b}$ for unknown $\underline{z}$ has $p^{\frac{n}{p}}$ solutions.

In C_p^n every element has order p so $w = \sum_{i=0}^{2^n-1} \alpha_i g_i \in \mathbb{Z}_p C_p^n$ satisfies $w^p = \sum_{i=0}^{2^n-1} \alpha$. When $w^p \neq 0$ then $w^p = \epsilon(w) \neq 0$ and the inverse of w is easy to obtain.

Thus in $\mathbb{Z}_p C_p^n$:

- Random matrices X may be chosen such that $X^p = 0$ and so X has large kernel.

- Random matrices A may be chosen such that $A^p = \alpha I$ for a scalar α and hence the inverse of A is easily obtained.
- It is possible to randomly choose X, Y so that $X^p = 0, Y^p = 0$ and so X, Y have large kernel and $(X + Y + 1)^p$ or $(X + Y + H)^p$ with $h \in C_p^n$ or $(X + Y + W)^p$ for various $w \in C_p^n$ to have value αI for a scalar α.
- If $\underline{x}$ is any vector considered in $\mathbb{Z}_p C_p^n$ then either $X^p = 0$ or else $(X + H)^p = I$ for any $h \in C_p^n$; more generally $X^p = 0$ or $(X + W)^p = 0$ for $\epsilon(w) = -\epsilon(\underline{x})$.

A generalised Walsh-Hadamard matrix $WH(p^n)$ is defined as follows. $WH(p) = F_p$ where F_p is the Fourier $p \times p$ matrix and $WH(p^n) = WH(p) \otimes WH(p^{n-1})$ for $n \geq 2$ where $\otimes$ denotes tensor product. This diagonalises any RC_p^n-matrix when the Fourier matrix exists. Using generalised Walsh-Hadamard matrices computations in $\mathbb{Z}_p C_p^n$ can be done in $O(m \log m)$ time, $m = p^n$.

8.5.3 With circulants

Suppose circulant matrices derived from $\mathbb{Z}_2 C_n$ are used where $n = 2m$ is large. Let C_n be generated by a. Let $J \subset \{0, 1, \ldots, m\}$ be chosen randomly. Now $w = (\sum_{j \in J}(a^j + a^{m+j})) + a^m$ satisfies $w^2 = (\sum_{j \in J}(a^{2j} + a^{2m+2j})) + a^{2m} = (\sum_{j \in J}(a^{2j} + a^{2j})) + 1 = 1$. (One could also use $w = (\sum_{j \in J}(a^j + a^{m+j})) + 1$.) The circulant matrix W which is the completion of w satisfies $W^2 = 1$. The number of choices for such J is of order 2^m. Then A, B above can then be constructed from choices of J.

Consider $\mathbb{Z}_p C_{pn}$ where C_{pn} is generated by a. It is easy to build singular elements w with $w^p = 0$. Now $(a^i + (p-1)a^{i+n})^p = a^i + (p-1)a^i = 0$ and also $(a^i + a^{i+n} + \ldots + a^{i+(p-1)n})^p = 0$ and other similar constructions. Taking a sum of such types gives an element w with $w^p = 0$ whose completion is a singular element with $\dim \ker \geq \frac{n}{p}$.

Matrix and vector multiplication for circulant matrices (RG-matrices for G cyclic) can be done with fast Fourier transform and so can be done in $O(n \log n)$ time.

Thus in $\mathbb{Z}_p C_{pm}$ random matrices may be chosen as follows:

- Random matrices X such that $X^p = 0$ and so X has large kernel.
- Random matrices A such that $A^p = \alpha I$ for a scalar α and hence the inverse of A is easily obtained.
- It is possible to randomly choose X, Y so that $X^p = 0, Y^p = 0$ and so X, Y have large kernel and $(X + Y + 1)^p$ or $(X + Y + H)^p$ with $h \in C_p^n$ or $(X + Y + W)^p$ for various $w \in C_p^n$ to have value αI for a scalar α.

Given data $\underline{x} = (\alpha_0, \alpha_1, \ldots, \alpha_{m-1})$ of length m we need this to be considered in a cyclic group ring so that its completion is singular of large kernel.

Let $\alpha_i \in \mathbb{Z}_2$. Consider the group ring $\mathbb{Z}_2 C_{2m}$ where g generates C_{2m} and let $\underline{\underline{x}} = \sum_{i=0}^{m-1} \alpha_i g^i + \sum_{i=0}^{2m-1} \alpha_i g^{i+m}$. (Yes, g^i and g^{i+m} have the same coefficient.) Then $\underline{\underline{x}}^2 = 0$ and clearly $\underline{x}$ is embedded in $\underline{\underline{x}}$ and the completion of $\underline{\underline{x}}$ has large kernel.

Let $\alpha_i \in \mathbb{Z}_p$. Consider the group ring $\mathbb{Z}_p C_{pm}$ and suppose C_{pm} is generated by g. Consider $\underline{\underline{x}} = \sum_{i=0}^{m-1} \alpha_i g^i - \sum_{i=0}^{m-1} \alpha_i g^{i+p}$. Then $\underline{\underline{x}}^p = 0$ and $\underline{x}$ is embedded in $\underline{\underline{x}}$.

8.5.4 Achieving properties for matrices of general group rings

For properties of group rings and related algebra consult [10]. The augmentation mapping $\epsilon : RG \to R$ is the ring homomorphism given by $\epsilon(\sum_{g \in G} \alpha_g g) = \sum_{g \in G} \alpha_g$. Let R be a field Suppose now w is non-singular. Then $\epsilon(w)$ is a unit of F and so is non-zero. Then $w' = w - \epsilon(w)1_g$ or $w' = w - \epsilon(w)g$ for any $g \in G$ satisfies $\epsilon(w') = 0$ and so w' is singular. Then W' is singular.

Let $\underline{x_1}$ be $1 \times r$ data considered as an element of a group ring FH where F is a field. If a key has already been exchanged there is no need to make the pieces of data singular.

Let G be an $r \times n$ generator matrix of a zero-divisor (n, r) code over FH. Then by Proposition 8.2 the completion X of $\underline{x} = \underline{x_1}G$ has rank at most r. Thus $\dim\ker X \geq (n-r)$. Provided r is not very large then given large n it is impossible to deduce X from AX or XA for an unknown (reasonable) matrix A. For example the code could have large rate say $\frac{3}{4}$ and then $\dim\ker X \geq \frac{n}{4}$; for n large then also $\dim\ker X$ is large. This is one way to ensure the data to be transmitted has large kernel and at the same time enabling error-correcting.

Thus if $\underline{x}$ is data to be transmitted considered as an element of the group ring RG then $\underline{x} - \epsilon(\underline{x})$ is always a singular element. However this element may have large rank. If this way of ensuring the data to be transmitted is singular is used then multiple vector design should be used. The data is broken as $(\underline{x_1}, \underline{x_2}, \ldots, \underline{x_r})$. Then its augmentation is added to each $\underline{x_i}$ to get a vector $\underline{y_i} = (\underline{x_i}, \epsilon(\underline{x_i}))$ which is then used. So for example $(\underline{y_1}A_1, \underline{y_2}A_2, \ldots, \underline{y_r}A_r)$ would be transmitted. Each piece is singular and r is large.

8.6 Complete orthogonal sets of idempotents

Here we consider properties of complete sets of idempotent matrices and ranks of the idempotents. These are used to construct X, Y such that these have large kernels and linear combinations of which are non-singular.

Let R be a ring with identity $1_R = 1$. A *complete family of orthogonal idempotents* is a set $\{e_1, e_2, \ldots, e_k\}$ in R such that
(i) $e_i \neq 0$ and $e_i^2 = e_i$, $1 \leq i \leq k$;
(ii) If $i \neq j$ then $e_i e_j = 0$;
(iii) $1 = e_1 + e_2 + \ldots + e_k$.

The idempotent e_i is said to be *primitive* if it cannot be written as $e_i = e_i^{'} + e_i^{''}$ where $e_i^{'}, e_i^{''}$ are idempotents such that $e_i^{'} \neq 0, e_i^{''} \neq 0$ and $e_i^{'} e_i^{''} = 0$. A set of idempotents is said to be *primitive* if each idempotent in the set is primitive.

For example such sets always exist in FG, the group ring over a field F, when $charF \not| \, |G|$; these idempotent sets are related to the representation theory of FG, see [10]. General methods for constructing such sets are derived in [6] and the reader is referred therein for details. The constructions in [6] were derived in connection with applications to multi-dimensional paraunitary matrices which are used in the communications' areas. Specific examples of large sets and using modular arithmetic (working over $GF(p)$) and where convolution methods may be applied are given in [7] .

For completeness some of the basics are given below.

Lemma 8.3 *Suppose $\{E_1, E_2, \ldots, E_s\}$ is a set of orthogonal idempotent matrices. Then* $\operatorname{rank}(E_1 + E_2 + \ldots + E_s) = tr\,(E_1 + E_2 + \ldots + E_s) = tr\,E_1 + tr\,E_2 + \ldots + tr\,E_s = \operatorname{rank} E_1 + \operatorname{rank} E_2 + \ldots + \operatorname{rank} E_s$.

Proof: It is known that $\operatorname{rank} A = \operatorname{tr} A$ for an idempotent matrix, and so $\operatorname{rank} E_i = \operatorname{tr} E_i$ for each i. If $\{E, F, G\}$ is a set an orthogonal idempotent matrices so is $\{E + F, G\}$. From this it follows that $\operatorname{rank}(E_1 + E_2 + \ldots + E_s) = \operatorname{tr}(E_1 + E_2 + \ldots E_s) = \operatorname{tr} E_1 + \operatorname{tr} E_2 + \ldots + \operatorname{tr} E_s = \operatorname{rank} E_1 + \operatorname{rank} E_2 + \ldots \operatorname{rank} E_s$. $\square$

Corollary 8.2 $\operatorname{rank}(E_{i_1} + E_{i_2} + \ldots + E_{i_k}) = \operatorname{rank} E_{i_1} + \operatorname{rank} E_{i_2} + \ldots + \operatorname{rank} E_{i_k}$ *for $i_j \in \{1, 2, \ldots, s\}$, and $i_j \neq i_l$ for $j \neq l$.*

Let $\{e_1, e_2, \ldots, e_k\}$ be a complete orthogonal set of idempotents in a vector space over F.

Lemma 8.4 *Let $w = \alpha_1 e_1 + \alpha_2 e_2 + \ldots + \alpha_k e_k$ with $\alpha_i \in F$. Then w is invertible if and only if each $\alpha_i \neq 0$ and in this case $w^{-1} = {\alpha_1}^{-1} e_1 + {\alpha_2}^{-1} e_2 + \ldots + {\alpha_k}^{-1} e_k$.*

Proof: Suppose each $\alpha_i \neq 0$. Then $w({\alpha_0}^{-1} e_0 + {\alpha_1}^{-1} e_1 + \ldots + {\alpha_k}^{-1} e_k) = e_0^2 + e_1^2 + \ldots + e_k^2 = e_0 + e_1 + \ldots + e_k = 1$.

Suppose on the other hand w is invertible and that some $\alpha_i = 0$. Then $we_i = 0$ and so w is a (non-zero) zero-divisor and is not invertible. $\square$

Now specialise the e_i to be $n \times n$ matrices and in this case use capital letters and let $e_i = E_i$.

Lemma 8.5 *Let $\{E_1, E_2, \ldots, E_k\}$ be a complete orthogonal set of idempotents in $F_{n\times n}$ and define $A = a_1E_1 + a_2E_2 + \ldots + a_kE_k$. Then A is invertible if and only if each $a_i \neq 0$ and in this case $A^{-1} = {a_1}^{-1}E_1 + {a_2}^{-1}E_2 + \ldots + {a_k}^{-1}E_k$.*

The reader may consult [6] for a proof of the following.

Proposition 8.1 *Suppose $\{E_1, E_2, \ldots, E_k\}$ is a complete symmetric orthogonal set of idempotents in $F_{n\times n}$. Let $A = a_1E_1 + a_2E_2 + \ldots + a_kE_k$ with $a_i \in F$. Then the determinant of A is $|A| = a_1^{\operatorname{rank} E_1} a_2^{\operatorname{rank} E_2} \ldots a_k^{\operatorname{rank} E_k}$.*

Lemma 8.6 *Let $\{E_0, E_1, E_2, \ldots, E_{n-1}\}$ be a complete orthogonal set of idempotents in $F_{n\times n}$ where each E_i has $\operatorname{rank} 1$. Let $I = \{0, 1, \ldots, n-1\}$ and $J \subset I$. Define $X = \sum_{j\in J} \alpha_j E_j$ with $\alpha_j \neq 0$. Then $\operatorname{rank} X = |J|$.*

Proof: Let $W = X + \sum_{j\in(I-J)} E_j$. Then by Lemma 8.5 W is invertible and so has $\operatorname{rank} n$. Hence $n = \operatorname{rank}(W) = \operatorname{rank}(X + \sum_{j\in(I-J)} E_j) \leq \operatorname{rank} X + \operatorname{rank} \sum_{j\in(I-J)} E_j = \operatorname{rank} X + (n - |J|)$, by Corollary 8.2. Therefore $\operatorname{rank} X \geq |J|$. From the rank inequality $\operatorname{rank}(AB) \geq \operatorname{rank} A + \operatorname{rank} B - n$, get $0 \geq \operatorname{rank} X + \operatorname{rank} \sum_{j\in(I-J)} E_j - n = \operatorname{rank} X + n - |J| - n$ and hence $|J| \geq \operatorname{rank} X$. Thus $\operatorname{rank} X = |J|$ □

The following Lemma may be proved similarly.

Lemma 8.7 *Let $\{E_0, E_1, E_2, \ldots, E_k\}$ be a complete orthogonal set of idempotents in $F_{n\times n}$ where $\operatorname{rank} E_i = r_i$. Let $I = \{0, 1, \ldots, k\}$ and $J \subset I$. Define $X = \sum_{j\in J} \alpha_j E_j$ with $\alpha_j \neq 0$. Then $\operatorname{rank} X = \sum_{j\in J} \operatorname{rank} E_j$.*

This enables the construction of public keys as follows.

Let $\{E_0, E_1, \ldots, E_{n-1}\}$ be a complete orthogonal set of idempotents in $F_{n\times n}$ and $I = \{0, 1, \ldots, n-1\}$.

1. A chooses $J \subset I$ with $|J|$ approximately half of $|I| = n$ and constructs $X = \sum_{j\in J} \alpha_j E_j$, $Y = \sum_{j\in(I-J)} \beta_j E_J$ with $\alpha_j \neq 0, \beta_j \neq 0$. It is enough to choose J so that both X, Y have large kernel.
2. A chooses $\{A_1, A_2\}$ non-singular and calculates $\{XA_1, YA_2\}$.
3. A has public key (XA_1, YA_2) and private key (X, Y, A_1, A_2).

More generally, proceed as follows to construct a public key for A.

Let $\{E_0, E_1, \ldots, E_k\}$ be a complete orthogonal set of idempotents in $F_{n\times n}$ and $I = \{0, 1, \ldots, k\}$.

1. A chooses $J \subset I$ and constructs $X = \sum_{j\in J} \alpha_j E_j$, $Y = \sum_{j\in(I-J)} \beta_j E_j$ with $\alpha_j \neq 0, \beta_j \neq 0$. Then $\operatorname{rank} X = \sum_{j\in J} \operatorname{rank} E_j$ and $\operatorname{rank} Y = \sum_{j\in(I-J)} E_J = n - \operatorname{rank} X$ and J needs to be chosen so that both X, Y have large kernel.
2. A chooses $\{A_1, A_2\}$ non-singular and calculates $\{XA_1, YA_2\}$.
3. A has public key (XA_1, YA_2) and private key (X, Y, A_1, A_2).

When B wishes to communicate $\underline{z}$ to A, the process is as follows.

1. B transmits $\underline{z}XA_1, \underline{z}YA_2$.
2. A works out $\underline{z}X, \underline{z}Y$ and then $\underline{z}(X + Y)$.
3. Now $X + Y$ is invertible by Lemma 8.5 and easy to calculate and A works out $\underline{z}$.

For each n there are many different complete orthogonal sets of idempotents in $F_{n\times n}$. It is not necessary that the particular set used by A in constructing her public key be known to the world so in fact an additional step (before 1.) in constructing public key could be:

0. A chooses a complete orthogonal set of idempotents $\{E_0, E_1, \ldots, E_k\}$ in $F_{n\times n}$.

Schemes where X, Y obtained from orthogonal sets of idempotents as above are 'protected' on both sides, as explained in section 4, may also be implemented; details are omitted.

8.7 Convolution

Let $\underline{z} * \underline{w}$ denote the (circulant) convolution of $\underline{z}$ and $\underline{w}$. Let A be a circulant matrix with first row $\underline{a}$.

Lemma 8.8 $\underline{x}A = \underline{x} * \underline{a}$.

Proof: Let X be the completion of $\underline{x}$. Then XA is a circulant matrix whose first row is $\underline{x}A$ and is also $\underline{x} * \underline{a}$. $\square$

More general G convolutions may be defined as follows. Let $\underline{x} \in R^n, \underline{y} \in R^n$ and G a finite group of order n. Define the G-convolution of $\underline{x}$ and $\underline{y}$, denoted $\underline{x} *_G \underline{y}$, as follows. Suppose $\underline{x} = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1}), \underline{y} = (\beta_0, \beta_1, \ldots, \beta_{n-1})$ with $\alpha_i, \beta_i \in R$ and $G = \{g_0, g_1, \ldots, g_{n-1}\}$. Let $x = \alpha_0 g_0 + \alpha_1 g_1 + \ldots + \alpha_{n-1} g_{n-1}, y = \beta_0 g_0 + \beta_1 g_1 + \ldots + \beta_{n-1} g_{n-1}$. Then $x \in RG, y \in RG$ and $xy = \gamma_0 g_0 + \gamma_1 g_1 + \ldots + \gamma_{n-1} g_{n-1}$ for some $\gamma_i \in R$. Define $\underline{x} *_G \underline{y} = (\gamma_0, \gamma_1, \ldots, \gamma_{n-1})$.

Lemma 8.9 *Let A be an RG-matrix with first row $\underline{a}$ and $\underline{x} \in R^n$. Then $\underline{x}A = \underline{x} *_G \underline{a}$.*

Proof: Let X be the completion of $\underline{x}$ in RG. Then XA is an RG-matrix whose first row is both $\underline{x}A$ and $\underline{x} *_G \underline{a}$. $\square$

When G is the cyclic group generated by g, with listing $\{1, g, g^2, \ldots, g^{n-1}\}$, then $\underline{z} *_G \underline{w}$ is the normal (circulant) convolution. Calculations in the cyclic group ring and with circulant matrices may be performed in $O(n \log n)$ time using a fast Fourier transform (FFT) and FTs allow an effective parallel implementation.

The encryption methods of the previous sections which involve multiplying vectors and matrices can be done in $O(n \log n)$ time when the matrices have a structure such as the structure of certain group ring matrices.

8.8 Coding aspects theory

Suppose data x_1 of size $1 \times r$ is to be transmitted. Encode x_1 by $x_1 G = \underline{x}$ where G is $r \times n$ generator matrix of an (n, r) code with G of rank r. If G is an $n \times n$ circulant matrix of rank r then the first r rows of G are linearly independent; this follows from for the following:

Lemma 8.10 *Let G_1 be a circulant $n \times n$ matrix of* rank r *and suppose G consists of the first r rows of G_1. Let $x = x_1 G$ where x_1 is a vector of size $1 \times r$ and let X be the completion of x. Then* $\operatorname{rank} X \leq r$.

Proof: Let $x_1 = (\alpha_1, \alpha_2, \ldots, \alpha_r)$. Then $x = x_1 G = (\alpha_1, \alpha_2, \ldots, \alpha_r, 0, 0, \ldots, 0) G_1$ where there are $(n-r)$ zeros. Then $X = \Gamma G_1$ where Γ is the completion of $(\alpha_1, \alpha_2, \ldots, \alpha_r, 0, 0, \ldots, 0)$. Hence $\operatorname{rank} X \leq \operatorname{rank} G_1 = r$ as required. $\square$

Lemma 8.11 *Let G be the generator $r \times n$ matrix of a cyclic zero-divisor (n, r) code and $x = x_1 G$ where x_1 has size $1 \times r$. Then the completion X of x has* rank $\leq r$.

Proof: Let the rows of G be denoted by $\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_r\}$. Then $x_1 G = \sum_{i=1}^{r} \alpha_i \hat{v}_i$. Let $\hat{G}$ be the circulant matrix from which G is derived and let the rows of this be denoted by $v_1, v_2, \ldots, v_n$. The first r rows of $\hat{G}$ are linearly independent, see [8], and thus $\hat{v}_i = \sum_{i=1}^{r} \beta_i v_i$ for some β_i. Hence $x = x_1 G = \sum \gamma_i v_i$ for some γ_i. Thus $x = x_1G = (\gamma_1, \gamma_2, \dots, \gamma_r, 0, 0, \dots, 0)\hat{G}$ where $\underline{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_r, 0, 0, \dots, 0)$ has length n. Hence $X = \Gamma\hat{G}$ where X is the completion of x and Γ is the completion of $\underline{\gamma}$. As $\operatorname{rank}\hat{G} = r$ this implies that $\operatorname{rank} X \leq r$. □

More generally we obtain the following result.

Proposition 8.2 *Let G be a rank r generator $r \times n$ matrix of a zero-divisor (n, r) code obtained from a group ring $n \times n$ matrix $\hat{G}$ of* rank r*. Then the completion of x_1G in this group ring has* rank $\leq r$.

Proof: Let the rows of G be $\{v_1, v_2, \dots, v_r\}$ and the rows of $\hat{G}$ be $\{w_1, w_2, \dots, w_n\}$. Now $\hat{G}$ has rank r and let $\{w_j | y \in J\}$ for $J \subset \{1, 2, \dots, n\}$ be a set of r linearly independent rows of $\hat{G}$.

Now $x_1G = \sum_{i=1}^{r} \alpha_i v_i$ and $v_i = \sum_{j \in J} \beta_{i,j} w_j$. Hence $x_1G = \sum_{j \in J} \delta_j w_j$. Define for $i = 1, 2, \dots, n$, $\gamma_i = \delta_i$ when $i \in J$ and $\gamma_i = 0$ when $i \notin J$. Then $x_1G = (\gamma_1, \gamma_2, \dots, \gamma_n)\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = (\gamma_1, \gamma_2, \dots, \gamma_n)\hat{G}$. Hence $X = \Gamma\hat{G}$ where X is the completion of x_1G and Γ is the completion of $(\gamma_1, \gamma_2, \dots, \gamma_n)$. As $\operatorname{rank}\hat{G} = r$ this implies $\operatorname{rank} X \leq r$. □

References

[1] U. Baum, "Existence and efficient construction of fast Fourier transforms on supersolvable groups", Comput. Complexity, 1/3, 235-256, 1994.

[2] Richard E. Blahut, *Algebraic Codes for data transmission*, Cambridge University Press, 2003.

[3] Neal Koblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

[4] Neal Koblitz, *Algebraic aspects of cryptography*, Springer, 2004.

[5] Ted Hurley, "Group rings and rings of matrices", Inter. J. Pure & Appl. Math., 31, no.3, 2006, 319-335.

[6] Barry Hurley and Ted Hurley, "Paraunitary matrices", arXiv:1205.0703.

[7] Barry Hurley and Ted Hurley, "Systems of MDS codes from units and idempotents", arXiv:1301.5596.

[8] Paul Hurley and Ted Hurley, "Codes from zero-divisors and units in group rings", Int. J. Inform. and Coding Theory, 1, 57-87, 2009.

[9] Alfred Menezes, Paul Van Oorschot and Scott Vanstone, *A Handbook of Applied Cryptography*, CRC Press, 2001.

[10] César Milies & Sudarshan Sehgal, *An introduction to Group Rings*, Klumer, 2002.