Formalization of Binary Fields and $N$-dimensional Binary Vector Spaces Using the Mizar Proof Checker

Kenichi Arai$^1$ and Hiroyuki Okazaki$^2$

$^1$Tokyo University of Science, 2641 Yamazaki Noda-City, Chiba 278-8510, Japan
$^2$Shinshu University, 4-17-1 Wakasato Nagano-city, Nagano 380-8553, Japan

Abstract—Binary fields and $n$-dimensional binary vector spaces play important roles in practical computer science, for example, coding theory and cryptology. In this paper, we introduce our formalization of binary fields and $n$-dimensional binary vector spaces. We then prove some theorems about subspaces and bases of $n$-dimensional binary vector spaces. We prove the correctness of our formalization using the Mizar proof checking system as a formal verification tool. Mizar is a project that formalizes mathematics with a computer-aided proving technique and is a universally accepted proof checking system. The main objective of this study is to prove the security of cryptographic systems using the Mizar proof checker.

Keywords: Formal Verification, Proof Checker, Mizar, Binary Field, $N$-dimensional Binary Vector Space

1. Introduction

Mizar[1] is a project that formalizes mathematics with a computer-aided proving technique. The objective of this study is to prove the security of cryptographic systems using the Mizar proof checker. To this end, we intend to formalize several topics concerning cryptology. As a part of this effort, we have previously introduced our formalization of the Advanced Encryption Standard (AES) at FCS’12[2].

The text that formalizes and describes the proof of mathematics by Mizar is called an “article”. When an article is newly described, it is possible to advance it by referring to articles registered in the MML that have already been inspected as proof. Likewise, other articles can refer to an article after it has been registered in the MML. Although the Mizar language is based on a descriptive method for general mathematical proofs, a reader should consult the references for its grammatical details because Mizar uses a specific, unique notation[5], [6], [7], [8].

2. Mizar

Mizar[1] is an advanced project of the Mizar Society led by A.Trybulec, which formalizes mathematics with a computer-aided proving technique. The Mizar project describes mathematical proofs in the Mizar language, which was created to formally describe mathematics. The Mizar proof checker operates in both Windows and UNIX environments, and registers the proven definitions and theorems in the Mizar Mathematical Library (MML). Mizar is one of the proof assistants that can mechanically check proofs written in the Mizar language.

3. Outline of Binary Fields and $N$-dimensional Binary Vector Spaces

In this section, we review binary fields, $n$-dimensional binary vector spaces, and vector subspaces[3].

3.1 Binary Fields

The binary field $\mathbb{F}_2$ (a so-called Galois field and written GF(2)) is a finite field with two elements: 0 and 1. The operations defined over the binary field $\mathbb{F}_2$ are binary addition and multiplication. Binary addition “+” and multiplication “•” are defined by the rules of modulo-2 arithmetic, as shown in Figure 1.
There are a total of $F^n$ vectors. The binary field $\mathbb{F}_2$ is a field, many of the familiar properties of number systems such as rational numbers and real numbers are retained; these include associativity, commutativity, and distributivity.

### 3.2 N-dimensional Binary Vector Spaces

The vector space $V$ consists of a set of elements over which the binary addition operation, denoted by the XOR (exclusive OR) and AND operations, respectively. Because $\mathbb{F}_2$ is a field, many of the familiar properties of number systems such as rational numbers and real numbers are retained; these include associativity, commutativity, and distributivity.

#### Figure 1: Binary Addition and Binary Multiplication

Here, $0$ is the additive identity and $1$ is the multiplicative identity. Binary addition and multiplication correspond to the XOR (exclusive OR) and AND operations, respectively. Because $\mathbb{F}_2$ is a field, many of the familiar properties of number systems such as rational numbers and real numbers are retained; these include associativity, commutativity, and distributivity.

### 3.3 Vector Subspaces

A subset $S$ of the vector space $V$ is called a subspace of the vector space $V$. A non-empty subset $S$ of $V$ is a subspace if it satisfies the following conditions:

1. $V$ is a commutative group for the binary addition operation.
2. For any $x \in \mathbb{F}$ and any $u, v \in V$, $x \cdot u = u = u \cdot x$.
3. For any $u, v \in \mathbb{F}$ and any $x$, $x \cdot (u + v) = x \cdot u + x \cdot v$.
4. For any $u \in V$ and any $y \in \mathbb{F}$, $(x + y) \cdot u = x \cdot u + y \cdot u$.
5. If $1$ is the unit element in $\mathbb{F}$, then $1 \cdot u = u$ for any $u \in V$.

Here, the elements of $V$ and $\mathbb{F}$ are called vectors and scalars, respectively.

Consider an ordered sequence of $n$ components $(x_1, x_2, \ldots, x_n)$ where each component $x_i$ is an element of the binary field $\mathbb{F}_2$. This sequence is called an $n$-component vector. There are a total of $2^n$ vectors. The corresponding vector space for this set of vectors is denoted as $V_n$, a vector space of dimension $n$.

The binary addition operation $\oplus$ for this vector space is defined as follows: if $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$ are vectors in $V_n$, then

$$u \oplus v = (u_1 \oplus v_1, u_2 \oplus v_2, \ldots, u_n \oplus v_n).$$

Since the sum vector is also an $n$-component vector, this vector also belongs to the vector space $V_n$, and so the vector space is said to be closed under the addition operation $\oplus$.

The addition of any two vectors of a given vector space is also another vector of the same vector space.

Furthermore, $V_n$ is a commutative group under the addition operation. The all-zero vector $0 = (0, 0, \ldots, 0)$ is also in the vector space and is the identity for the addition operation:

$$u \oplus 0 = (u_1 \oplus 0, u_2 \oplus 0, \ldots, u_n \oplus 0) = u,$$

$$u \oplus u = (u_1 \oplus u_1, u_2 \oplus u_2, \ldots, u_n \oplus u_n) = 0.$$

Each vector of a vector space defined over the binary field is its own additive inverse. It can be shown that the vector space defined over $\mathbb{F}_2$ is a commutative group, so that associative and commutative laws are satisfied. The multiplication between a vector of the vectorial space $u \in V$ and a scalar of the binary field $x \in \mathbb{F}_2$ can be defined as

$$x \cdot u = (x \cdot u_1, x \cdot u_2, \ldots, x \cdot u_n).$$

It can be shown that addition and scalar multiplication obey the associative, commutative, and distributive laws, so the set of vectors $V_n$ is a vector space defined over the binary field $\mathbb{F}_2$. In addition, a vector space over $\mathbb{F}_2$ is a binary vector space; therefore, it is called an $n$-dimensional binary vector space $V_n$.

### 3.3 Vector Subspaces

A subset $S$ of the vector space $V$ is called a subspace of the vector space $V$. A non-empty subset $S$ of $V$ is a subspace if it satisfies the following conditions:

- For any two vectors in $S$, $u, v \in S$, the sum vector $(u + v) \in S$.
- For any element of the field $x \in \mathbb{F}$ and any vector $u \in S$, the scalar multiplication $x \cdot u \in S$.

If $\{v_1, v_2, \ldots, v_k\}$ is a set of vectors of the vector space $V$ defined over $\mathbb{F}$ and $x_1, x_2, \ldots, x_k$ are scalar numbers of the field $\mathbb{F}$, the sum $x_1 \cdot v_1 + x_2 \cdot v_2 + \cdots + x_k \cdot v_k$ is called a linear combination of the vectors $\{v_1, v_2, \ldots, v_k\}$.

A set of $k$ vectors $\{v_1, v_2, \ldots, v_k\}$ is said to be linearly dependent if and only if there exist $k$ scalars of $\mathbb{F}$, not all equal to zero, such that a linear combination is equal to the all-zero vector:

$$x_1 \cdot v_1 + x_2 \cdot v_2 + \cdots + x_k \cdot v_k = 0.$$

If the set of vectors is not linearly dependent, then this set is said to be linearly independent.

A set of vectors is said to generate (span) a vector space $V$ if each vector in that vector space is a linear combination of the vectors of the set. In any vector space or subspace, there exists a set of at least linearly independent vectors that generate such a vectorial space or subspace.

For a given $n$-dimensional binary vector space $V_n$, the set of vectors $\{e_1, e_2, \ldots, e_n\} =$
{(1,0,\ldots,0),(0,1,\ldots,0),\ldots,(0,0,\ldots,1)} \text{ is the set of vectors } e_i \text{ that has a non-zero component only at position } i. \text{ This set of vectors is linearly independent.}

This set of linearly independent vectors \( \{e_1, e_2, \ldots, e_n\} \) generates the \( n \)-dimensional binary vector space \( V_n \), whose dimension is \( n \) and is called a basis of the \( n \)-dimensional binary vector space \( V_n \). If \( k < n \), the set of linearly independent vectors \( \{v_1, v_2, \ldots, v_k\} \) generates \( S \) of the \( n \)-dimensional binary vector space \( V_n \) through all their possible linear combinations:

\[
c = y_1 \cdot v_1 \oplus y_2 \cdot v_2 \oplus \cdots \oplus y_k \cdot v_k.
\]

The subspace formed is of dimension \( k \) and consists of \( 2^k \) vectors. The number of combinations is \( 2^k \) because the coefficients \( y_i \in \mathbb{F}_2 \) adopt only one of the following two possible values: 0 or 1.

4. Binary Fields and their Algebraic Structures

In this section, we formalize the binary field \( \mathbb{F}_2 \) as an algebraic structure. First, we define “BinaryField” as a structure that has BOOLEAN as its carrier and that includes the following two binary operations on BOOLEAN: “XORB1” and “MLB1.” Next, we prove that BinaryField has the attributes of a field.

We define binary addition and multiplication as follows:

```
definition func XORB1 -> BinOp of BOOLEAN
  means for x,y be Element of BOOLEAN
  holds it.(x,y) = x 'xor' y;
end;
```

```
definition func MLB1 -> BinOp of BOOLEAN
  means for x,y be Element of BOOLEAN
  holds it.(x,y) = x '&' y;
end;
```

“BOOLEAN” denotes the binary set \{0, 1\} in the Mizar language. Here, each of the operations, XORB1 and MLB1, is defined as a “BinOp of BOOLEAN,” which is a function from (BOOLEAN, BOOLEAN) into BOOLEAN[9], [10]. Thus, the definitions of both XORB1 and MLB1 ensure that their outputs are elements of BOOLEAN.

We can now define an algebraic structure that has BOOLEAN as its carrier, two binary operations, and two identity elements:

```
definition func BinaryField -> strict non empty doubleLoopStr
  equals doubleLoopStr(# BOOLEAN,XORB1,MLB1, TRUE,FALSE #);
end;
```

Here, BinaryField has TRUE (1) and FALSE (0) as the identities of MLB1 and XORB1, respectively.

In the Mizar language, “Field” denotes a field as follows[11]:

```
definition mode Field is add-associative
  right_zeroed right_complementable
  Abelian commutative associative
  well-unital distributive
  almost_left_invertible non
generated doubleLoopStr;
end;
```

Data types are constructed as “modes” in the Mizar language. We can construct new modes in the Mizar language using existing modes and attributes, which are characteristics of modes and other objects.

Now, we explain attributes of “doubleLoopStr” \((F,f,g,e_f,e_f)\), where \( F \) is BOOLEAN, \( f \) is XORB1, \( g \) is MLB1, \( e_f \) is TRUE, and \( e_f \) is FALSE. Here, “add-associative right_zeroed ... almost_left_invertible non
degenerated” are attributes. The attribute “add-associative” means that \((F,f)\) holds the associative law:

\[
\forall a,b,c \in F, f(f(a,b),c) = f(a,f(b,c)).
\]

The attribute “right_zeroed” means that \( e_f \) is an additive identity in \( F \), that is, \( \forall a \in F, f(a,e_f) = a \). The attribute “right_complementable” means that for any \( a \in F \), there exists an additive inverse \( b \) of \( a \) in \( F \), that is, \( f(a,b) = e_f \).

The attribute “Abelian” means that \((F,f)\) holds the commutative law: \( \forall a,b \in F, f(a,b) = f(b,a) \). The attribute “commutative” means that \((F,g)\) holds the commutative law: \( \forall a,b \in F, g(a,b) = g(b,a) \). The attribute “associative” means that \((F,g)\) holds the associative law:

\[
\forall a,b,c \in F, g(g(a,b),c) = g(a,g(b,c)).
\]

The attribute “well-unital” means that \( e_g \) is a multiplicative identity in \( F \), that is, \( \forall a \in F, g(a,e_g) = g(e_g,a) = a \).

\(^1\) An algebraic structure “doubleLoopStr” is a set for which two binary operations and two elements are defined.
The attribute “distributive” means that \((F, f, g)\) holds the distributive law:

\[
\forall a, b, c \in F, \quad g(a, f(b, c)) = f(g(a, b), g(a, c)),
\]
\[
\forall a, b, c \in F, \quad g(f(a, b), c) = f(g(a, c), g(b, c)).
\]

The attribute “almost_left_invertible” means that for any \(a \in F \setminus \{0\}\), there exists a multiplicative inverse \(b\) of \(a\) in \(F \setminus \{0\}\), that is, \(g(b, a) = e_g\). The attribute “non degenerated” means that there exists non-zero divisor \(a(\neq 0) \in F\), where \(g(a, b) = 0\) for some element \(b(\neq 0) \in F\). Then, if “doubleLoopStr” \((F, f, g, e_f, e_l)\) has all the above attributes, the algebraic structure is a field.

Next, we introduce the following “cluster” for Binary-Field:

```
registration
cluster BinaryField ->
  add-associative right_zeroed
  right_complementable Abelian
  commutative associative
  well-unital distributive
  almost_left_invertible
  non degenerated;
end;
```

This cluster is equivalent to the following theorem:

```
theorem
  BinaryField is add-associative
    right_zeroed right_complementable
    Abelian commutative associative
    well-unital distributive
    almost_left_invertible
    non degenerated doubleLoopStr;
```

In the Mizar system, a cluster captures some properties of an expression, for example, its attributes. Once a cluster has been registered, these properties can be derived automatically from the expression, although it is always necessary to refer to theorems.

Finally, we prove the following theorem:

```
theorem
  BinaryField is Field;
```

### 5. N-dimensional Binary Vector Spaces and their Algebraic Structures

In this section, we formalize the \(n\)-dimensional binary vector space \(V_n\) as an algebraic structure. We then prove some theorems about subspaces and bases of \(n\)-dimensional binary vector spaces.

#### 5.1 Formalization of \(N\)-dimensional Binary Vector Spaces

In this section, we explain the definition of a vector space \(V\) as an algebraic structure that has already been formalized in Mizar. Then, we define “\(n\)-BinaryVectSp” as the structure that has \(n\)-tuples on BOOLEAN as its carrier and that includes addition and scalar multiplication on \(n\)-tuples on BOOLEAN, “XORB” and “MLTB”.

In Mizar, the vector space algebraic structure, “VectSpStr over \(F\),” has already been formalized as follows[11]:

```
definition
  let F be 1-sorted;
  struct(addLoopStr) VectSpStr over F
    (# carrier -> set,
      addF -> BinOp of the carrier,
      ZeroF -> Element of the carrier,
      lmult ->
        Function of
          [:the carrier of F,
            the carrier:],
          the carrier #);
  end;
```

Mizar supports multiple inheritance of structures, making a whole hierarchy of interrelated structures available in the MML. In that hierarchy, the “1-sorted” structure is the common ancestor of almost all other structures[12].

The definition of a vector space, “VectSp of \(F\),” has already been formalized as follows[11]:

```
definition
  let F be add-associative
    right_zeroed right_complementable
    Abelian associative well-unital
    distributive non empty
    doubleLoopStr;
  mode VectSp of F is
    vector-distributive
    scalar-distributive
    scalar-associative
    scalar-unital
    add-associative right_zeroed
    right_complementable Abelian
    non empty VectSpStr over F;
end;
```

Here, “vector-distributive,” “scalar-distributive,” “scalar-associative,” and “scalar-unital” are attributes and correspond
to (iii), (iv), (v), and (vi) of Section 3.2, respectively. This definition also satisfies (i) from VectSpStr over $F$. See the Appendix for details of this definition. If VectSpStr over $F$ has all the above attributes, the algebraic structure is a vector space $V$. Thus, we will use VectSpStr over $F$ and VectSp of $F$ to formalize the $n$-dimensional binary vector space $V_n$. “$n$-BinaryVectSp.”

We define addition and scalar multiplication on $n$-tuples on BOOLEAN, “XORB $n$” and “MLTB $n$,” as follows:

```
definition let n be non zero Element of NAT; func XORB n -> BinOp of n-tuples_on BOOLEAN means for x,y being Element of n-tuples_on BOOLEAN holds it.(x,y) = Op-XOR(x,y); end;
```

```
definition let n be non zero Element of NAT; func MLTB n -> Function of [:the carrier of BinaryField, n-tuples_on BOOLEAN:], n-tuples_on BOOLEAN means for a be Element of BOOLEAN, x be Element of n-tuples_on BOOLEAN, i be set st i in Seg n holds (it.(a,x)).i = a '&' x.i; end;
```

“NAT” denotes the set of natural numbers with 0 in the Mizar language. Here, Op-XOR is a bitwise XOR function and Seg $n = [1, n]$.

The additive identity, the all-zero vector, for $n$-tuples on BOOLEAN is defined as follows:

```
definition let n be non zero Element of NAT; func ZeroB n -> Element of n-tuples_on BOOLEAN equals n |-> 0; end;
```

Here, $n |-> 0$ is the all-zero $n$-tuples $(0,0,...,0)$.

The following theorem can then be proved:

```
theorem VectSpStr(# n-tuples_on BOOLEAN, XORB n,ZeroB n,MLTB n #) is VectSp of BinaryField;
```

We now define an algebraic structure that has $n$-tuples on BOOLEAN as its carrier, addition, scalar multiplication, and the additive identity, as follows:

```
definition let n be non zero Element of NAT; func n-BinaryVectSp -> VectSp of BinaryField equals VectSpStr(# n-tuples_on BOOLEAN, XORB n,ZeroB n,MLTB n #);
```

```
5.2 Theorems about Subspaces and Bases of $N$-dimensional Binary Vector Spaces
```

In this section, we prove some theorems about subspaces and bases of $n$-dimensional binary vector spaces. Definitions of linear independence, bases, dimension, and subspace have already been formalized in Mizar[13], [14], [15]. Therefore, we use those definitions to prove some theorems.

First, we formalize the theorem about linear independence as follows:

```
definition let n be non zero Element of NAT; func n-BinaryVectSp is VectSp of BinaryField;
```

Finally, we prove the following theorem:

```
theorem for n,m be non zero Element of NAT, A be FinSequence of n-tuples_on BOOLEAN, B be finite Subset of n-BinaryVectSp st rng A = B & m <= n & len A = m & A is one-to-one & (for i,j be Nat st i in Seg n & j in Seg m holds (i = j implies (A.i).j = TRUE) & (i <> j implies (A.i).j = FALSE)) holds B is linearly-independent;
```
Here, rng A of A that holds \( "\text{(for } i,j \text{ be Nat st } i \in \text{Seg } n \& \ldots \& (i < j \text{ implies } (A.i).j = \text{FALSE})" \) is a basis of n-BinaryVectSp. Because this rng A is linearly independent, any subset B of rng A is also linearly independent.

Second, we formalize the following theorem that relates to the above theorem:

\[
\text{theorem}
\begin{align*}
\text{for } n \text{ be non zero Element of } \text{NAT}, \\
A \text{ be FinSequence of } \\
\text{n-tuples on BOOLEAN,} \\
B \text{ be finite Subset of } n\text{-BinaryVectSp} \\
\text{st } \text{rng } A = B \& \text{len } A = n \& \\
A \text{ is one-to-one } & \\
\text{for } i,j \text{ be Nat st } i \in \text{Seg } n & \\
& j \in \text{Seg } n \text{ holds} & \\
& (i = j \text{ implies } (A.i).j = \text{TRUE}) \& & \\
& (i < j \text{ implies } (A.i).j = \text{FALSE}) \\
\text{holds } \text{Lin } B = \text{VectSpStr} & \\
\text{(# the carrier of n-BinaryVectSp,} & \\
\text{the addF of n-BinaryVectSp,} & \\
\text{the ZeroF of n-BinaryVectSp,} & \\
\text{the lmult of n-BinaryVectSp } \\ & \\
\end{align*}
\]

Here, Lin B means the space generated (spanned) by B. The space generated by B has an algebraic structure equal to n-BinaryVectSp.

We can then formalize the following theorem about the basis:

\[
\text{theorem}
\begin{align*}
\text{for } n \text{ be non zero Element of } \text{NAT}, \\
A \text{ be FinSequence of } \\
\text{n-tuples on BOOLEAN,} \\
B \text{ be finite Subset of } n\text{-BinaryVectSp} \\
\text{st } \text{rng } A = B \& \text{len } A = n \& \\
A \text{ is one-to-one } & \\
\text{for } i,j \text{ be Nat st } i \in \text{Seg } n & \\
& j \in \text{Seg } n \text{ holds} & \\
& (i = j \text{ implies } (A.i).j = \text{TRUE}) \& & \\
& (i < j \text{ implies } (A.i).j = \text{FALSE}) \\
\text{holds } \text{Lin } B = \text{VectSpStr} & \\
\text{(# the carrier of n-BinaryVectSp,} & \\
\text{the addF of n-BinaryVectSp,} & \\
\text{the ZeroF of n-BinaryVectSp,} & \\
\text{the lmult of n-BinaryVectSp } \\ & \\
\end{align*}
\]

If B is a basis of n-BinaryVectSp, this theorem means that B, which is a finite subset of n-BinaryVectSp, exists.

Next, we formalize a theorem about dimension as follows:

\[
\text{theorem}
\begin{align*}
\text{for } n \text{ be non zero Element of } \text{NAT}, \\
A \text{ be FinSequence of } \\
\text{n-tuples on BOOLEAN,} \\
C \text{ be Subset of } n\text{-BinaryVectSp} \\
\text{st } \text{len } A = n \& A \text{ is one-to-one } & \\
\text{card } (\text{rng } A) = n \& & \\
\text{for } i,j \text{ be Nat st } i \in \text{Seg } n & \\
& j \in \text{Seg } n \text{ holds} & \\
& (i = j \text{ implies } (A.i).j = \text{TRUE}) \& & \\
& (i < j \text{ implies } (A.i).j = \text{FALSE}) & \\
& C \text{ c= } \text{rng } A \text{ holds} & \\
\text{Lin } C \text{ is Subspace of } n\text{-BinaryVectSp} & \\
& C \text{ is Basis of } \text{Lin } C & \\
\text{dim } (\text{Lin } C) = \text{card } C; \\
\end{align*}
\]

Here, c= means . This theorem means that the space generated by C, which is a subset of rng A, is a subspace of n-BinaryVectSp. In that case, C is a basis of the space generated by C and the dimension of Lin C is equal to the cardinal number of C.

6. Conclusion

In this paper, we introduced our formalization of binary fields, n-dimensional binary vector spaces, and their algebraic structures in Mizar. We also proved some theorems about subspaces and bases of the n-dimensional binary vector spaces using the Mizar proof checking system as a formal verification tool. Currently, we are analyzing the cryptographic systems using our formalization in order to achieve the security proof of cryptographic systems.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers 21240001 and 22300285. We would like to sincerely thank Prof. Shidama at Shinshu University for his helpful advice.
References


Appendix

(ii) is formalized as follows:

```haskell
definition let F be non empty 1-sorted, 
V be non empty VectSpStr over F; 
let x be Element of F; 
let v be Element of V; 
func x*v -> Element of V 
equals (the lmult of V).(x,v); 
end;
```

(iii) is formalized as follows:

```haskell
definition let F be non empty doubleLoopStr; 
let IT be non empty VectSpStr over F; 
attr IT is vector-distributive 
means 
for x,y being Element of F 
for v being Element of IT 
holds x*(y+v) = x*y+x*v; 
end;
```

(iv), (v), and (vi) are formalized as follows:

```haskell
definition let F be non empty doubleLoopStr; 
let IT be non empty VectSpStr over F; 
attr IT is scalar-distributive 
means 
for x,y being Element of F 
for v being Element of IT 
holds (x+y)*v = x*v+y*v; 
end;
```

Moreover, the elements of *V* and *F* are called vectors and scalars, respectively; they are formalized as follows:

```haskell
definition let F be 1-sorted; 
let VS be VectSpStr over F; 
mode Vector of VS is Element of VS; 
mode Scalar of F is Element of F; 
end;
```