InterNational Committee for Information Technology Standards
INCITS Secretariat, Information Technology Industry Council (ITI)
1250 Eye St. NW, Washington, DC 20005
Telephone 202-737-8888; Fax 202-638-4922
Email: incits@itic.org

DOC: M1/05-0841
Date: December 02, 2005

Bioscrypt Technical Contribution towards Project 1749D

Conformance Testing Methodology Standard, Part 3 – Finger Pattern Conformance Testing Methodology

Project editor contact information:

Omid S. Jahromi
Bioscrypt Inc.
Phone: (905) 940-7776
Email: omid.jahromi@bioscrypt.com

1 Introduction

Recently, INCITS M1 has developed a number of biometric data interchange format standards for different biometric modalities. The main goal of these standards is to provide a means for different biometric systems from different vendors to share data. They would also allow components of a biometric system to be substituted with other components from different vendors with minimum effort.

From a user's point of view, it is vital that systems claiming agreement with a standard actually meet the standards criteria and are fully interoperable. It's also important, from the point of view of standards development, to test whether it is indeed possible to assure interoperability between various vendors by implementing a standard's technical requirements. Two major problems with base standards that may only become apparent during conformance testing are the following:

- some requirements may be undefined so that the specification of these areas is left to each vendor
- some requirements may be ambiguous so that there is a chance for misinterpretation caused by the wording of the base standard

In view of the above considerations, there is a need for the development of a systematic conformance testing methodology. Unfortunately, when it comes to verifying conformance to a given standard, no test can provide a conformance proof[1]. A well-designed test can, however, identify the most likely sources of problems and ensure that the system under test conforms to the standard under a reasonably broad set of circumstances, giving assurance, but not a guarantee, of conformance.

This document describes a two-step compliance test with the ANSI-INSITS 377-2004 "Finger Pattern Data Interchange Format" standard. It provides some general guidelines for conformance testing and then defines the specific assertions that must be checked in each step.

2 Acknowledgment

This contribution has been prepared using some of the ideas and methodologies put forward by Dr. John Campbell in *M1/05-0650 – "NBSP Technical Contribution towards "Conformance Testing Methodology Standard for Biometric Data Interchange Format Standards - Part 1 - Generalized Conformance Testing Methodology"*. The Editor would like to acknowledge Dr. Campbell's efforts and contributions towards the development of viable conformable testing standards.

---

[1] Please see Section 6 for a detailed explanation why this is the case.

3 Scope

This document specifies only those concepts, test assertions and test methodologies that pertain to ANSI-INSITS standard 377-2004 “Finger Pattern Data Interchange Format”. It does not cover conformance testing for CBEFF data structures embedded in the above standard.

4 Normative References

The following normative references are critical to the proper understanding and application of this standard.

INCITS 377-2004, Finger Pattern Based Interchange Format

5 Terms and Definitions

5.1 **Base standard** - The standard containing the specification that is the subject of the conformance testing

5.2 **Biometric Data Interchange Record (BDIR) –** A data package containing biometric data that claims to be in the form prescribed by the base standard

5.3 **Conformance –** Fulfillment by a product, software, or data interchange file of all relevant base standard requirements

5.4 **Conformance testing laboratory –** Organization that carries out conformance testing

5.5 **Conformance Test Suite (CTS) –** A test software which is used to automate certain tasks in conformance testing

5.6 **Implementation Under Test (IUT) –** A product, a computer algorithm or a data interchange record (file) which implements the base standard being tested.

6 The Nature of Conformance Testing

The INCITS 377-2004 “Finger Pattern Based Interchange Format” standard, as its name implies, establishes a set of guidelines and requirements for a data interchange format that can be used for pattern-based verification or identification of fingerprint images.

The goal of having a standard data interchange format is to assure the end users that a BDIR produced by any conformant product can be correctly read and interpreted by any other conformant product. There are thus two types of fundamental conformance claims:

- **TYPE A**: the ability to **produce** conformant BDIRs
- **TYPE B**: the ability to **use** conformant BDIRs

In general, given a BDIR, it is only possible to prove that the system that produced it is *non-conformant* to the standard. This is because, due to lack of knowledge about the internal elements of the system producing the BDIR and the raw biometric sample which was used by that system, many of the requirements of the standard cannot be checked. On the other hand, even one instance of contradiction while inspecting the BDIR is sufficient to conclusively refute the conformance of its producing system. Consequently, every genuine test regarding "TYPE A" claims is logically an attempt to falsify the claim. In other words, the conformance test reduces to a search for one or more counter examples[2].

In this first draft version of our contribution only "TYPE A" conformance is considered. Testing "TYPE B" is left as an open subject for subsequent versions.

7 Testing "TYPE A" conformance claims

STEP I: A first step towards the goal of refuting the conformance of a BDIR is checking to see if all of the specified fields and data structures in the BDIR are consistent with the base standard's requirements. At this stage we are not concerned about the fidelity of the information contained in the BDIR.

The BDIR produced by the IUT ***shall*** be checked for field-by-field and byte-by-byte conformance with the specification of the data interchange format in the base standard. This must be done both in terms of fields included and the ranges of the values in those fields.

Furthermore, the test laboratory ***shall*** describe in the conformance report the detailed structure of the BDIRs tested. Specifically, the presence or absence of optional fields and the value of variable fields (e.g., number of views) ***shall*** be listed. The total number of BDIRs tested and the input finger print images used to create those BDIRs ***shall*** be reported for each structural variant of the BDIR. Note that in some cases, multiple input images (containing single views of the same finger) may have had to be combined to produce a single BDIR. The number of live biometric characteristics involved in generating each structural variant of the BDIRs (either through live capture and BDIR generation by an IUT or through the previous generation of a fingerprint image library) ***shall*** be reported.

It is important to follow the above procedure because an IUT may have the capability to produce multiple BDIRs, depending on the requirements of the application in which it is used. Some of these BDIRs may be conformant while others may not. Thus, it is important to specify which types were tested and how many of each type. In an ideal

---

[2] This is a particular instance of the infamous "demarcation test" which distinguishes scientific theories from non-scientific ones. Every genuine scientific theory is *prohibitive*, in the sense that it forbids, by implication, particular events or occurrences. As such it can be tested and falsified, but never logically verified. See Karl Popper, *Conjectures and Refutations: The Growth of Scientific Knowledge*, Routledge, London, 1963.

world every possible combination of parameters for a particular data interchange format would be tested, but this is not realistic given the resources that would be required for such testing. By reporting the presence or absence of optional fields and the values for variable structural fields, it is possible for an end user of the base standard to determine if the particular variant of the base standard tested is appropriate for their needs. The end user may also require conformance certification for a specific type of BDIR. An obvious example would be a two finger BDIR, since many applications require enrollment of more than one biometric characteristic in order to allow for a back-up if one of them becomes damaged or temporarily unusable.

STEP II: The second step in testing "TYPE A" conformance of an IUT is checking the internal consistency of the BDIRs produced by it. This step of the test essentially relates values from one part of the BDIR to values from other parts of the BDIR. The specific assertions tested ***shall*** be those described in Annex A of this contribution.

8 Assertion Elements for Steps I and II testing

In this section we will focus on specifying a set of assertions that must be verified by the conformance testing laboratory in order to declare the IUT conformant.

For each row of the table in Annex A, a particular condition defined by a test assertion shall be checked and the BDIR is determined to have either passed or failed the test assertion. If a BDIR passes all assertions in both STEP I and II of conformance testing specified in Annex A, then the BDIR ***shall*** be declared conformant. A minimum of 100 BDIRs representing at least 25 different fingers ***shall*** be tested in order to declare and IUT conformant. If even a single BDIR is not conformant, then the testing laboratory ***shall*** report that the IUT was not conformant.

8.1 Operators

To determine conformance of a data interchange record we compare the value of each field with a quantity or range of quantities which are known to be either valid or invalid according to explicit or implicit requirements of the base standard. These quantities may be determined in advance (e.g. Format Identifier), or calculated during the test from context dependant data within the BDIR (e.g. Length of Record). A list of basic math operators used during this process is given below.

- **Equal (EQ)**
- **Not-equal (NEQ)**
- **Greater than or equal (GTE)**
- **Less than or equal (LTE)**
- **Greater than (GT)**
- **Less than (LT)**
- **In range (In)**

  Indicates that the test ***shall*** pass if the field value is within the specified range

- **Incremental (INC)**

  Indicates the test ***shall*** pass if the field value is in sequence and within the specified range relative to the last instance of this field within the current data set. This includes ensuring that the value of the first field instance is at the start of the specified range. (e.g. View Number)

- **Calculation (C)**

  Indicates the test ***shall*** pass if the field value meets a certain criteria that cannot be simply expressed by one of the other operations (e.g. unit conversion from 1/100th mm to pixels). The necessary calculations are described in a note following the table.

8.2 Operands

Numerical operand values are expressed in decimal (e.g. 73) or hexadecimal (e.g. 0x49) notation. A range of values are expressed by listing the lower bound, followed by the word "to", followed by the upper bound (e.g. 1 to 255). Where a test requires more than one operand, values and ranges are separated by a comma. A very simple mathematical calculation, involving a number and a **Field Name** or a pair of **Field Names** may be expressed directly as an operand. For instance, 1 - ({Y Size of Finger Pattern} - {Y Cellular Offset}) indicates a range from a minimum of 1 to a maximum of the size of the finger pattern in the y-direction minus the cellular offset in the y-direction.

- **{Field Name}**

  When referring to a value stored within a particular field, the tables use the **Field Name** surrounded by braces, e.g. {Number of Views}.

- **Read**

  Refers to the number of data subsets within the BDIR which contain the data associated with a particular group of related elements defined in the base standard. The Read operand is always given in conjunction with a descriptive name that explains which data subsets it refers to from the base standard. This value is recorded by the conformance testing software when reading the BDIR. The particular data subsets read are context dependent, but examples would include Finger Views Read and Minutiae Read.

- **Bytes Read**

Refers to the number of bytes within a specific subset of the BDIR which contain the data associated with a group of related elements The Bytes Read operand is always used in conjunction with a field which refers to the byte length of a subset of data from the base standard. This value is recorded by the conformance testing software when reading the BDIR. Total Bytes Read Refers to the total number of bytes within the BDIR, as recorded by the conformance testing software when reading the BDIR.

- **Bytes Expected**

Refers to the total number of bytes expected (calculated from the appropriate fields) within a specific subset of the BDIR which contains the data associated with a particular group of related elements defined in the base standard The Bytes Expected operand is always used in conjunction with a field which refers to the byte length of a subset of data from the base standard. The calculation required for computing the Bytes Expected will be provided in a note following the table.

- **Total Bytes Expected**

Refers to the total number of bytes expected (calculated from the appropriate fields) within the BDIR

8.3 Other assertion elements

To be added in subsequent versions.

Annex A – Step I and II assertions to be checked for INCITS 377-2004 conformance testing

Test	Field	Operator	Operands	References	Step
1	CBEFF Format Type	EQ	0x0301, 0x0302	6.1	1
2	Format Identifier	EQ	0x46505200	6.2.1	1
3	Version Number	EQ	0x20313000	6.2.2	1
4	Length of Record	IN	44 to $(2^{32} -1)$	6.1, 6.2.3	1
4.1	Length of Record	EQ	Total bytes present in the record	6.1, 6.2.3	2
4.2	Length of Record	EQ	Total Bytes Expected (Note 1)	6.1, 6.2.3	2
5	CBEFF PID Owner	NEQ	0	6.2.4	1
6	CBEFF PID Type		All values are valid. Thus, this filed is not tested.	6.2.4	
7	Number of Finger Patterns	IN	1 to 255	Table 3	1
7.1	Number of Finger Patterns	EQ	Patterns present in the record	Table 3	2
8	X Size of Finger Pattern	IN	1 to 255	Table 3	1
9	Y Size of Finger Pattern	IN	1 to 255	Table 3	1
10	X Resolution of Finger Pattern	IN	1 to 788	Table 3	1
11	Y Resolution of Finger Pattern	IN	1 to 788	Table 3	1
12	X Number of Cells	C	(Note 2)	Table 3	2
13	Y Number of Cells	C	(Note 3)	Table 3	2
14	X Number of Pixels in Cells	IN	1 to ({X Size of Finger Pattern} - {X Cellular Offset})	Table 3	2
15	Y Number of Pixels in Cells	IN	1 to ({Y Size of Finger Pattern} - {Y Cellular Offset})	Table 3	2
16	X Cellular Offset	IN	0 to {X Size of Finger Pattern}	Table 3	2
17	Y Cellular Offset	IN	0 to {Y Size of Finger Pattern}	Table 3	2
18	Bit-depth of Cell Structure Angle	IN	1 to 8	Table 3	1
19	Bit-depth of Cell Structure Wavelength	IN	1 to 8	Table 3	1
20	Bit-depth of Cell Structure Phase Offset	IN	1 to 8	Table 3	1
21	Bit-depth of Cell Structure Quality	IN	1 to 8	Table 3	1
22	Cell Quality Granularity	IN	1 to 8	Table 3	1
23	Reserved Bytes	EQ	0	6.2.21	1
24	Finger Location	IN	0 to10	6.3.1.1	1
25	Impression Type	IN	0 to 3, 8, 9	6.3.1.2	1

26	Number of Views in Finger Pattern	IN	1 to 255	Table 3	1
27	Finger Pattern Quality	IN	-2, -1, 0 to 100	6.3.1.4	1
28	Length of Data Block	EQ	Bytes present in the record	6.3.1.5	2
28.1	Length of Data Block	EQ	Bytes Expected (Note 4)	6.3.1.5	2
29	View Number	INC	0 to {Number of Views -1}	6.3.2.1.1	2
30	Cell Angle	NONE			
31	Cell Wavelength	NONE			
32	Cell Phase Offset	NONE			
33	Cell Group Quality	NONE			
34	Extended Data	NONE			

8.4 Notes

8.4.1 Note 1 (4.2) – Total Bytes Expected is calculated using the following algorithm.

```
SUMBYTES = 36

FOR I = 1 TO {Number of Finger Patterns}
   SUMBYTES = SUMBYTES + 6 + {Length of Data Block}
END

{Total Bytes Expected} = SUMBYTES
```

8.4.2 Note 2 (12) – Range of {X Number of Cells}

We assume that any pixels on the right side of the image which are less than {X Number of Pixels in Cells} will be ignored and that there will not be a smaller cell created at the edge of the image.

```
{X number of Cells} EQ 1 – FLOOR(({X Size of Finger Pattern} - {X Cellular Offset}) / {X Number of Pixels in Cells})
```

8.4.3 Note 3 (13) – Range of {Y Number of Cells}

We assume that any pixels on the bottom of the image which are less than {Y Number of Pixels in Cells} will be ignored and that there will not be a smaller cell created at the edge of the image.

```
{Y number of Cells} EQ 1 – FLOOR(({Y Size of Finger Pattern} - {Y Cellular Offset}) / {Y Number of Pixels in Cells})
```

8.4.4 Note 4 (28.1) – {Length of Data Block Bytes Expected}

If there is no extended data, i.e. if {CBEFF Format Type}=0x0301 then {Length of Data Block}= "Bytes Expected" for every finger pattern in the BIR. If, however, there is extended data (i.e. if {CBEFF Format Type}= 0x0302) then some finger patterns may have extended data and others may not and since the length of the extended data may vary from one pattern to the next within the same BIR. Thus, the only possible test is {Length of Data Block} GTE {Length of Data Block Bytes Expected}.

The following method of calculating {Length of Data Block Bytes Expected} assumes that the Finger Pattern Cell Data, Cell Quality Data and Finger Pattern Extended Data all begin at the start of a new byte, so there may be wasted bits within a byte at the end of each of these data sets. It also assumes that the cell quality groups are always square arrays of cells and that if there are insufficient pattern cells at the right edge or bottom edge of the pattern cell data to fit a complete cell quality group, then those cells don't have any cell quality assigned to them. For example, if the image is divided into 9 by 11 cells and {Cell Quality Granularity} = 3, then there will be 3 by 3 = 9 cell structure quality values.

```
NUMCELLS = {X Number of Cells} * {Y Number of Cells}

PATTERNBYTES = CEILING (NUMCELLS * ({Bit-depth of Cell Structure Angle} + {Bit-depth of Cell Structure Wavelength} + {Bit-depth of Cell Structure Phase Offset}) / 8)
```

```
QUALITYBYTES = CEILING(FLOOR({X Number of Cells} / { Cell Quality Granularity}) *
FLOOR({Y Number of Cells} / { Cell Quality Granularity}) * {Bit-depth of Cell Structure
Quality} / 8)
```

{Length of Data Block Bytes Expected} = 1 + PATTERNBYTES + QUALITYBYTES