ID3FINGER MOC 5.2

SPECIFICATIONS

Document number: SPC9T027

Revision 1.1

13.04.2016

Revision History

Revision	Date	Description
1.0	10.02.2016	Initial release
1.1	13.04.2016	Added accuracy indications

Contents

1 Introduction

1.1 Purpose

This document describes the API of the fingerprint on-card comparison implementation designed by id3 Technologies.

1.2 Terms and Definitions

Term	Definition
Algorithm	A sequence of instructions that tell a biometric system how to solve a particular problem. An algorithm will have a finite number of steps and is typically used by the biometric engine (i.e. the biometric system software) to compute whether a biometric sample and template match.
Biometric data	Data encoding a feature or features used in biometric verification.
Biometric information template	A constructed data object in a card containing information needed by the outside world for a verification process, see ISO/IEC 7816-11.
Comparison	The process of comparing a biometric sample with a previously stored reference template or templates.
Enrollment	The process of collecting biometric samples from a person and the subsequent preparation and storage of biometric reference templates representing that person's identity.
Extraction	The process of converting a captured biometric sample into biometric data so that it can be compared to a reference template; sometimes called "characterization".
Match / Matching	The process of comparing a biometric sample against a previously stored template and scoring the level of similarity.
Minutiae	Friction ridge characteristics that are used to individualize a fingerprint. Minutiae occur at points where a single friction ridge deviates from an uninterrupted flow. Deviation may take the form of ending, bifurcation, or a more complicated "composite" type.
Template / Reference template	Data, which represents the biometric measurement of an enrollee, used by a biometric system for comparison against subsequently submitted biometric samples. NOTE – this term is not restricted to mean only data used in any particular recognition method, such as template matching.

Table 2: Terms and Definitions

1.3 Abbreviated Terms

For the purpose of this specification, the following abbreviations apply.

BDB	Biometric Data Block
BDT	Biometric Data Template
BIT	Biometric Information Template as defined in ISO/IEC 7816-11
CBEFF	Common Biometric Exchange Formats Framework

DET	Detection-Error-Tradeoff (curve)
DO	Data Object
FMR	False Match Rate
FNMR	False Non Match Rate
MOC	Match-On-Card
PIV	Personal Identity Verification
RFU	Reserved for Future Use

1.4 References

Normative References

- ISO/IEC 7816-4:2005 – Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange
- ISO/IEC 7816-11:2004 – Identification cards – Integrated circuit cards – Part 11: Personal verification through biometric methods
- ISO/IEC 19794-2:2005 – Information technology – Biometric data interchange formats – Part 2: Finger minutiae data
- ISO/IEC 19785-1:2006 – Common biometric exchange formats framework – Part 1: Data element specification

2 Match-on-Card Algorithm

id3Finger MOC 5.2 is a finger minutiae comparison algorithm specifically designed and optimized for card-based systems. The algorithm is provided as a software library coded in C language and compiled on request for your specific secure IC chip.

2.1 Features

- Compliant with ISO/IEC 19794-2 compact card format
- High interoperability with most of the template generators available on the market
- AFIS-grade accuracy
- MINEX III compliant
- High tolerance to distortion and sensor variations
- Invariant to translation
- Configurable rotation tolerance up to 180°
- Suitable for small sensors
- Unique capability of score-level fusion for highly secure authentication using multiple fingers
- Low resource requirements
- Easy implementation on any target (C API, single library, no external dependencies)

2.2 Memory Requirements

Code Size

The memory footprint of the library (code size) is around 10 kB. This value can vary depending on the target processor.

Working Buffer

A working buffer must be allocated for the purpose of the fingerprint comparison process. The size of the buffer depends on the maximum number of minutiae that can be processed by the algorithm. This value is configurable on request.

The table below sums up the memory requirements depending on the maximum number of minutiae.

Number of minutiae	40 minutiae	60 minutiae	80 minutiae	128 minutiae
Template size	121 bytes	181 bytes	241 bytes	385 bytes
Working buffer size	729 bytes	789 bytes	849 bytes	993 bytes
Total size	**971 bytes**	**1151 bytes**	**1331 bytes**	**1763 bytes**

Table 4: Memory Requirements

NOTE – These values are computed on a 32-bit CPU and may slighlty vary on 16-bit or 64-bit CPUs.

2.3 Biometric Data Format

The data presented to the matcher shall comply with the ISO/IEC 19794-2 compact card format (ISO-CC). This format contains only the minutiae data with no header. Each minutia is coded using 3 bytes, as following:

- coordinate x (8 bits), unit = 10^{-1} mm
- coordinate y (8 bits), unit = 10^{-1} mm
- minutia type t (2 bits):
  - 00 = other,
  - 01 = ridge ending
  - 10 = ridge bifurcation
  - 11 = reserved for future use
- angle θ (6 bits), unit = $2\pi/64$

x-coordinate	y-coordinate	type t	angle θ
1 byte	1 byte	1 byte	

Notes

- The maximum value for the x and y coordinate is 25.5 mm with the compact format.
- Minutia (x, y, θ) triples must be unique.
- ISO 19794-2 compact size minutiae format encodes each coordinate on 8 bits, with one unit = 0.1 mm. As a result, fingerprint minutia must fit into an area of 2.55 x 2.55 cm^2. If the fingerprint scanner has a larger capture window, it is important that the captured coordinates be shifted to the upper left corner to ensure that no coordinates have an absolute value above 255 when transmitted to the card. Coordinate overflow is not authorized and may result in a false rejection. Coordinate overflow is not checked by the on-card matcher.
- The maximum number of minutiae that can be processed by the comparison algorithm is **128**. This value may be further reduced on request to save memory (see Memory Requirements for details). If the number of minutiae exceeds the maximum value, then truncation is necessary. In that case, the use of minutia quality values for removal is highly recommended.
- No minimum number of minutiae is required to perform a comparison but a minimum of 12 minutiae in both templates is generally recommended to get a positive match.
- No special ordering scheme is required for minutiae.

2.4 Decision Threshold

A decision threshold applied on the algorithm's output score determines the operational FMR (False Match Rate), i.e. the probability of a system to falsely accept fingerprints (impostors). Since FMR and FNMR (False Non-Match Rate) are in inverse proportion to each other, FNMR will increase with higher decision thresholds.

The decision threshold shall be defined according to the following table:

False Match Rate	Decision Threshold
1 %	2045
0.1 %	2828
0.01 %	3785
0.001 %	5011

False Match Rate	Decision Threshold
0.0001 %	6569
0.00001 %	8534

Table 5: Decision Threshold

NOTE – The values above also apply when the fusion algorithm is used.
This make it easier to implement multiple fingerprint comparisons on card.

For intermediate values at FMR values lower than $10e^{-2}$, the following Weibull model should be used:

$$threshold = \frac{ln(\frac{1}{1-FMR})^{-\beta} - b}{a}$$

Where:
a = 0.000551293622483652
b = 0.47
β = 0.1020
FMR <= $10e^{-2}$
ln is the natural logarithm

3 API Reference

This section provides the API reference for C programming language.

3.1 Type Definitions

The following types are C-definitions used in the native library.

```
#include <stdint.h>
typedef int8_t          int8;          // signed 8-bit integer
typedef int16_t         int16;         // signed 16-bit integer
typedef int32_t         int32;         // signed 32-bit integer
typedef uint8_t         uint8;         // unsigned 8-bit integer
typedef uint16_t        uint16;        // unsigned 16-bit integer
typedef uint32_t        uint32;        // unsigned 32-bit integer
```

3.2 Predefined Constants

The following constants are defined:

Name	Value	Description
ID3MOC_VENDOR_ID	003Fh	Identifier of the owner of the template matcher (value assigned by the IBIA)
ID3MOC_VERSION	0520h	Algorithm version number
ID3MOC_MIN_MINUTIAE	0	Minimum number of minutiae required
ID3MOC_MAX_MINUTIAE	128*	Maximum number of minutiae expected
ID3MOC_MINUTIA_ORDER	00h	Required ordering scheme for minutiae – No ordering required
ID3MOC_TEMPLATE_SIZE	385*	Maximum size (in bytes) of a biometric template encoded using a proprietary format.
ID3MOC_WORKING_BUFFER_SIZE	993*	Required size (in bytes) of the working buffer used for biometric comparison.
ID3MOC_FUSION_BUFFER_SIZE	24	Required size (in bytes) for score level fusion.

* Reducing the number of minutiae allow to lower the required size of the buffers used for biometric comparison. The following formulas apply:

- ID3MOC_TEMPLATE_SIZE = ID3MOC_MAX_MINUTIAE * 3 + 1
- ID3MOC_WORKING_BUFFER_SIZE = ID3MOC_MAX_MINUTIAE * 3 + 609

3.3 Usage

3.3.1 Enrolment

To prepare the input finger minutiae data for later verification, proceed as follow:

- Call the id3FingerMOC_Prepare function to create a reference template from an ISO-CC minutiae data,
- Store the reference template into static memory (EEPROM, Flash, etc.).

3.3.2 Verification

To perform a fingerprint comparison, proceed as follow:

- Retrieve the reference template from static memory (EEPROM, Flash, etc.) and copy to a buffer in RAM,
- Allocate memory for the working buffer,
- Call the id3FingerMOC_Prepare function to create the probe template,
- Call the id3FingerMOC_Compare function to perform the comparison,
- Analyze the output comparison score and make the final decision (match or no match) based on a decision threshold.

3.4 Functions

3.4.1 id3FingerMOC_Prepare Function

This function prepares a template to a further comparison process. The input biometric template shall comply with the ISO/IEC 19794-2 Compact Card format without appended tags and lengths. The output data is encoded in a proprietary format.

```
uint16 id3FingerMOC_Prepare(
        uint8 * isocc_template,
        uint16 template_size,
        uint8 * output_buffer );
```

Parameters

isocc_template
[in] Points to a buffer in RAM containing the reference biometric data in ISO/IEC 19794-2 compact card format.

template_size
[in] Supplies the length of the isocc_template parameter. Since each minutia is 3 bytes, the minutiae count is given by template_size / 3. The value shall be less than or equal to ID3MOC_MAX_MINUTIAE * 3.

output_buffer
[out] Points to a buffer in RAM that receives the prepared template. The output buffer must be at least ID3MOC_TEMPLATE_SIZE bytes long.

Returns

The size in bytes of the template transformed into the proprietary format.
Should equals ID3MOC_TEMPLATE_SIZE.

Remarks

For memory saving, the output buffer may start at the same address as the input buffer.

3.4.2 id3FingerMOC_Compare Function

This function performs the comparison between a probe template and a reference template.

```
uint16 id3FingerMOC_Compare(
        uint8 * working_buffer,
        uint8 * reference,
        uint8 * probe);
```

Parameters

working_buffer
[in] The working buffer used during the comparison process.
The required buffer size is ID3MOC_WORKING_BUFFER_SIZE bytes.

reference
[in] Points to a buffer in RAM that contains the reference template encoded using the proprietary format. Use the id3FingerMOC_Prepare function to prepare the reference template.

probe
[in] Points to a buffer in RAM that contains the probe template encoded using the proprietary format. Use the id3FingerMOC_Prepare function to prepare the probe template.

Returns

A similarity score value as a short integer in the range [0, 65535]. See Decision Threshold for details.

Remarks

Prior to a call to this function, both templates must be prepared using the id3FingerMOC_Prepare function.

3.4.3 id3FingerMOC_SetMaxRotation Function

This function sets the maximum rotation allowed between two fingerprint templates. Note that higher values may tend to increase comparison times.

```
void id3FingerMOC_SetMaxRotation(
        uint8 max_rotation );
```

Parameters

max_rotation
[in] The maximum rotation (unit = 1.40625°).

Remarks

Default value is 30 (i.e. 42°).
The maximum allowed rotation value is 127 (i.e. 180°), while minimum value is 12 (i.e. 17°).

3.4.4 id3FingerMOC_FuseScores Function

This function fuses the scores obtained by a single user for multi-finger comparison purposes.

```
uint16 id3FingerMOC_FuseScores(
        uint8* fusion_buffer,
        uint16* scores,
        uint8 count);
```

Parameters

fusion_buffer
[in] The buffer used during the fusion process. The required buffer size is ID3MOC_FUSION_BUFFER_SIZE.

scores
[in] An array containing scores to be fused.

count
[in] The number of scores to fuse.

Returns

A similarity score value as a short integer in the range [0, 65535]. See Decision Threshold for details.

Remarks

The number of fused scores should not exceed 10.
The thresholds to apply on fused scores are similar to those with single finger.

4 Implementation on a Smart Card

This section provides general information to implement fingerprint on-card comparison on a smart card.

4.1 APDU Commands

4.1.1 PUT DATA

The PUT DATA card command is used to store biometric reference data into the card.

CLA	'00' or '10' indicating command chaining)
INS	'DB'
P1	'3F'
P2	'FF'
Lc	Variable
Data Field	Biometric Data Template
Le	Absent

Table 7: PUT DATA APDU Command

4.1.2 VERIFY

The command initiates the comparison in the card of stored reference data with verification data sent from the interface device.

CLA	'00' or '10' indicating command chaining)
INS	'21'
P1	'00'
P2	'00'
Lc	Variable
Data Field	Biometric Data Template
Le	Absent

Table 8: VERIFY APDU Command

4.2 Data Objects

4.2.1 Biometric Data Template (BDT)

This object is sent to the card for the purpose of enrolment of the cardholder fingerprint as well as for verification of the cardholder fingerprint. This data object must be presented to the card as an ISO/IEC 19794-2 compact size Finger Minutiae format DO within a valid Biometric Data Template.

BDT is referred with tag '**7F2E**' and is defined as follows:

Tag	Len	Value
7F2E	Var.	**Biometric data template**

		Tag	Len	Value
		81	Var.	Finger minutiae data (ISO 19794-2 Compact size format)
		95	03.	Biometric type (1 byte fixed - '08') || Biometric subtype || Impression type

Table 9: Biometric Data Template

NOTE – id3Finger MOC 5.2 does not use additional features such as ridge count data, cores or deltas.

4.2.2 Biometric Information Template (BIT)

The Biometric Information Template (BIT) provides descriptive information regarding the associated biometric data.

Prior to a verification process, the BIT may be retrieved from a card to correctly compute and structure the biometric verification data (ie. number of minutiae, minutiae order, etc.).

Tag	Len	Value							
7F60	Var.	**Biometric Information Template (BIT)**							
		Tag	**Len**	**Value**					
		A1	Var.	**Biometric Header Template (BHT)**					
				Tag	**Len**	**Value**	**Description**		
				81	01	08	Biometric type (08 = fingerprint)		
				82	01	xx	Biometric subtype (finger position)		
				89	01	xx	Impression type		
				87	02	0101	CBEFF BDB format owner registered with IBIA (0101 = ISO/IEC JTC1/SC37)		
				88	02	0005	CBEFF BDB format type registered with IBIA (0005 = ISO/IEC 19794-2 Compact Card Format - Table 16)		
				B1	07	**Biometric matching algorithm parameters**			
						Tag	**Len**	**Value**	**Description**
						81	02	0080	Minimum and maximum numbers of minutiae accepted by the on-card matcher. See ISO/IEC 19794-2 (sub-clause 8.3.3 Table 10).
						82	01	00	Minutiae order. See ISO/IEC 19794-2 (sub-clause 8.3.4 Table 11 and 12)

Table 10: Biometric Information Template

4.2.3 Biometric Subtype

Biometric subtype (finger position) as per ISO/IEC 19785:

Hex value	Finger position
’00’	No information given
‘05’	Right thumb
‘09’	Right index
‘0D’	Right middle
‘11’	Right ring
‘15’	Right little
‘06’	Left thumb
‘0A’	Left index
‘0E’	Left middle
‘12’	Left ring
‘16’	Left little

Table 11: Finger position as defined in ISO/IEC 19785

4.2.4 Finger Impression Type

Impression type is used to prevent a false rejection with the on-card matcher when the enrolment template and the verification template have been captured using different types of sensor (e.g. flat scanner vs. swipe) because of associated distortions during fingerprint scanning. In such case, the same finger can be enrolled multiple times, once with each sensor and the on-card matcher shall perform the match of the verification template only with the enrolled template of the same type.

Hex value	Impression type
‘00’	Live-scan plain
‘01’	Live-scan rolled
‘02’	Nonlive-scan plain
‘03’	Nonlive-scan rolled
‘04’	Latent impression
‘05’	Latent tracing
‘06’	Latent photo
‘07’	Latent lift
‘08’	Swipe
‘09’	Vertical roll
‘18’	Live-scan optical contactless plain
‘1C’	Other
‘1D’	Unknown

Table 12: Finger Impression Type as defined in ISO/IEC 19785

5 Matching Performance

This section describes the matching performance of id3Finger MOC 5.2. All the figures are extracted from MINEX III Report.

5.1 Accuracy

The figure bellow shows the matching accuracy of the id3Finger MOC 5.2 as reported on NIST MINEX III results. Brown line shows the accuracy using id3 Extractor 2.5, while orange line shows the average accuracy using all other MINEX III compliant generators.

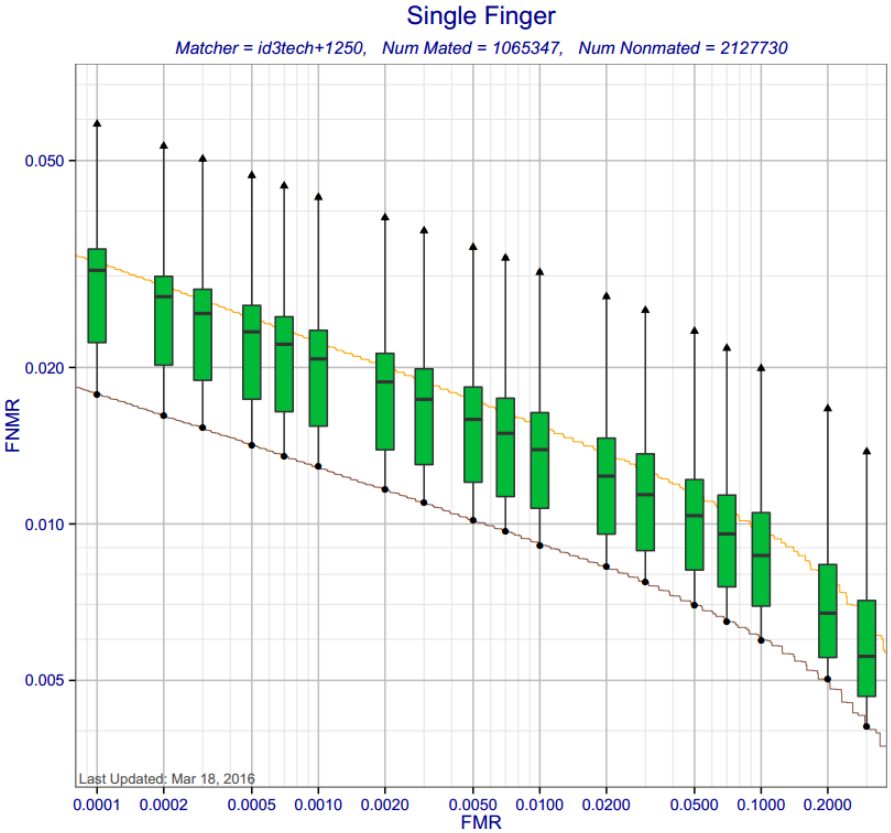

Figure 1: DET curves based on MINEX III results

5.2 Score Stability

As demonstrated by the MINEX III results, id3Finger MOC 5.2 is robust to spoofing and other active attacks, as the algorithm does not allow FMR to rise if the number of available minutia decreases nor increases (brown curves).

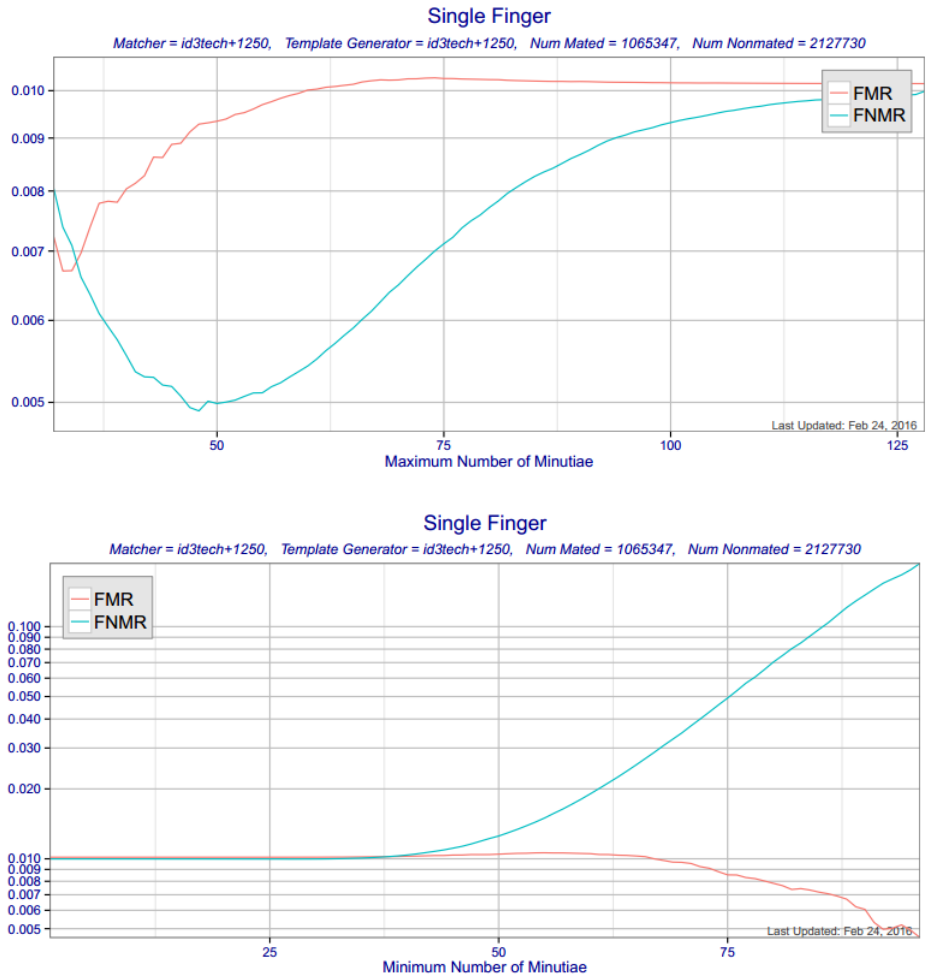

Figure 2: FNMR and FMR as a function of the maximum and minimum number of minutiae found by the template generators

6 Ordering Information

For any further information, please send an email to contact@id3.eu

id3 Technologies
5, rue de la Verrerie
38120 Le Fontanil-Cornillon
FRANCE

Tel: +33 (0)4 76 75 75 85
Fax: +33 (0)4 76 75 52 30

Internet: http://www.id3.eu
Contact: contact@id3.eu