Automatic Remote Evaluation System for Biometric Testing

Raul Sanchez-Reillo, Raul Alonso-Moreno
University Carlos III of Madrid
Univ. Group for Identification Technologies
Avda. Universidad, 30
E-28911 – Leganes (Madrid); SPAIN
{rsreillo, ramoreno}@ing.uc3m.es

Adam Czajka
Research and Academic Computer Network (NASK)
Biometrics Laboratory
Adam.Czajka@nask.pl

Young-Bin Kwon
Chung-Ang University
Computer Science and Engineering Dpt.
ybkwon@cau.ac.kr

***Abstract*—We present here a report on a new Performance Evaluation System for Biometric Systems, which is secure, automatic and remote. This system will provide developers in Biometrics to progress in their works, avoiding problems with data protection policies related to testing, but without compromising the privacy of testing subjects. Biometric testing needs personal data to be recorded and used. Therefore in order to test their prototypes, researchers have to waste a lot of time and money for buying or creating testing databases. The solution described here offers the developers a secure and remote system which is available at all times, and which concentrates all private data in a secured centralized server. Also, as current standards, including standard APIs, are used, efforts needed by developers to use the system will be minimized, lowering also the overhead costs for testing purposes. The system is described by block diagrams as well as flowcharts.**

Keywords—Application Program Interface (API), Biometrics, Evaluation, Personal Information Databases, Biometric Service Provider (BSP), Pre-processing, Feature Extraction, Verification.

I. Introduction to Personal Data and Biometric Evaluation

Along the last 5 years, concern about the level of privacy of personal data has highly increased. On the other hand, exchange and use of personal data has become a common practice in many environments. This has led to the development of Data Protection Laws in many countries, which are mandatory in all areas where information is used (such as [1]). Being satisfactory for citizens, these laws have showed up as a drawback for R&D community. There are many research areas where Personal Data is to be used as to improve the performance of the devices, products or algorithms developed.

Biometrics is one of such areas which have to deal with personal information (fingerprints, iris, face, signatures, etc.). In order to develop a biometric system, identification algorithms have to be tested using databases containing such personal information. Although the decision about whether biometric data is private or public is to be done, what everybody agrees is that it is personal and shall be protected. In fact, all Data Protection Laws developed which cover biometrics, have already demanded a strong protection for such data.

The following section will introduce the reader to the current situation and challenges related to data protection.

A. Data Protection

In a few words, the idea behind Data Protection Laws is that the citizen has to be aware about the use of his/her data, and has the right to withdraw such data at any moment. The awareness is solved usually by asking the citizens to sign a document where all the information is stated. But the withdrawal process is not trivial at all. The owner of the database has to delete such data from all databases, as well as from all the copies of such databases. When a centralized system is considered, it is possible to implement a mechanism removing that user from the current database, as well as from backup copies. But when such database has been distributed, no control is possible among copies, not being able to really know how many times and where such citizen's data is stored in computers worldwide. The reliability of solutions based on legal clauses included into the distribution contract has been proven to be limited.

Therefore, trying to cope with such laws, a system where personal data is being used, should ensure that:

- each of the citizens having his/her data in such system is

Raul Sanchez-Reillo, Raul Alonso-Moreno, Adam Czajka, Young-Bin Kwon, "Automatic Remote Evaluation System for Biometric Testing", 10th International Conference on Control, Automation, Robotics and Vision, ICARCV 2008, Hanoi, 17 - 20 December 2008

fully informed about what is such data collected for, and how it will be used,

- the system owner has a written authorization of each of such users, for storing and using his/her data,
- each user is aware of his right and knows the procedure how to withdraw his/her data from the system,
- non-authorized access to personal data is always denied (no matter the mean). In this case, not only direct access to data has to be avoided, but also indirect access by deriving information from reports issued,
- a restricted distribution policy, with strong implications to the clients where personal data is going to be sent,
- a fully operational procedure of the user withdrawal from the entire system, and all its copies.

B. Databases used in Biometric Identification Development

As mentioned earlier, biometrics needs databases with personal data, not only for evaluation, but also for technology development. The reliability of the results achieved when developing a biometric system, highly depends on the size, diversity and quality of the databases used. Each time a new version is developed, its performance has to be tested by developers to analyze the level of improvement acquired. If such tests are done with a small database of samples acquired in specific, fixed conditions and originating from a selected and uniform population, there is a danger that the system reliability is overestimated. Such a system placed in real world situations may fail in its performance.

Therefore, developers need large and diverse databases. In order to get them, two major alternatives exist. They can try to acquire them, from the several options (not many) available (eg. [2]-[10]). But new developments of biometric databases are facing serious problems in many countries in order to allow their distributions. Some have decided to emulate biometric samples, not being the real ones from the users (i.e., fake signatures, synthetic fingerprints [9], or synthetic irises [10]).

The other alternative that developers can face, is to build their own database, but building a biometric database is complicated and very time consuming task. If those developers need a really large database to better approximate populations existing in a real world, it may happen that the cost, in time and money, of building such database will overcome the one of the development of the entire system.

C. Current Evaluation Systems

Therefore there is a need to supply the R&D community with the tools demanded by them, such as good databases that would be free of any kind of legal issues. A very different approach is to apply for a position in one of the few public contest existing nowadays. Major examples of such contests are the ones related with fingerprint [11], face [4] or iris [3]. As being public contests, results are being made public (although some of them allow taking part anonymously), and take place in a certain period. These drawbacks make these evaluation only possible to those developers that already have a final product available, not being possible to use them during the development process of new algorithms or versions.

D. Challenge to Be Faced

With all this in mind, a new challenge shows up. How to provide the R&D community with the tools needed for high quality development of the biometric systems?. Focusing the challenge, is it possible to supply with the databases needed?. Authors have thought that the best way to face this challenge is to develop an Evaluation System that will be:

- centralized as to minimize all legal issues,
- fully secured, as to guarantee protection to all personal data,
- remotely accessible and performing evaluations automatically as to be available at all time for developers,
- able to hold all available databases,
- with a standardized interface to allow developers submitting their compiled solutions at minimum effort.

Covering all the above mentioned items, the paper describes the Evaluation System developed by the authors. Therefore, in the following section an introduction to BioAPI (the standard API for Biometrics) will be given. Then the general architecture of the Evaluation System will be presented, followed by the section covering its operational description. At the last stages of the paper, the requirements for developers will be given, as well as an outline of the results provided currently.

II. The Standard API for Biometric Interchange

If an evaluation system is to be developed, a common interface shall be developed as to allow the full interaction about such system with the developers using it. As compiled versions are to be used (not source code), and automatic evaluation is demanded, an Application Program Interface (API) is needed. Instead of developing their own API as it has happened in the past with the evaluation contests, authors propose the use of all standardized technology available.

Coming to a Biometric API, ISO/IEC JTC1/SC37 has developed the 19784-1 standard [12], most commonly known as BioAPI 2.0. This standard comes from the input of the BioAPI Consortium [13], which is an organization founded in 1998, with the target of building a multi-level API architecture. In March 1999, BioAPI Consortium joined efforts with the US Biometric Consortium, who had already developed a high-level biometric API. In March 2001, version 1.1 of the API, as well as its reference implementation was launched. As the Biometrics Subcommittee was formed by ISO/IEC JTC1, BioAPI

Consortium offered their work to be considered as a standard. During the standard development, some improvements where made in BioAPI, becoming version 2.0 - the one really standardized under the number 19784. This multipart standard is still under development and the aim is to issue new versions that could fit all the diversity of requirements of emerging biometric systems.

The basic idea of BioAPI is given in Fig. 1. It is based on a Framework that serves as an interface between the application and the biometric devices. In order to communicate with any external application, the Framework offers an API (set of functions) which work independently of how the devices are being developed.. On the other hand, in order to communicate with the devices, the Framework offers another API (here called SPI), with all the functions needed to access such devices. Independently of the device type, they shall provide a driver with the supported services, as to allow them to be accessed by the Framework. Such a driver is called Biometric Service Provider (BSP).

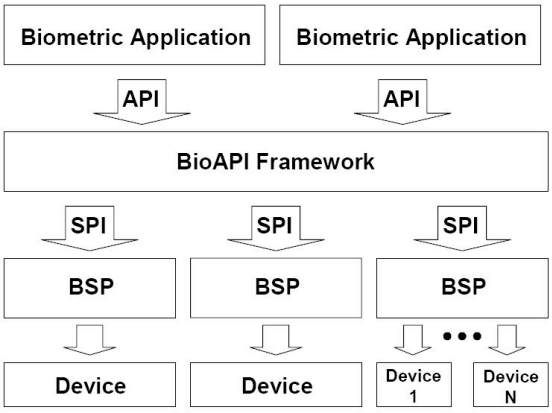

Fig. 1: BioAPI API/SPI Model [12]

BSPs can be nearly of any kind. They can be related to just one physical device, more than one, or just none (e.g. being an algorithm). Due to the functions provided by the SPI, BSPs can be dynamically loaded and unloaded from the Framework. Any external application can use any of the BSPs loaded through the Framework allowing any system complexity. On the other way, BSPs can even talk with the Framework to access other BSPs loaded. BSPs and their functionalities will be one of the features most used in the Evaluation System presented in this paper. Regarding functions to provide to the Framework, section V.1 gives a list of the ones needed by ARES.

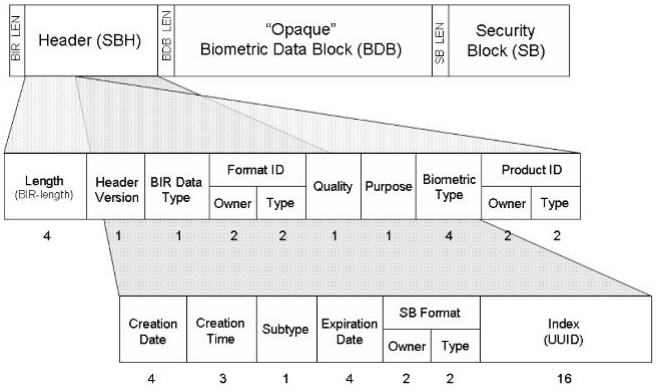

Fig. 2: BIR Format [12]

Section V will enter in detail with all requirements that developers will have to cover, as to be able to develop the BSPs needed to apply their algorithms to the Evaluation System.

Another important issue will be how data is exchanged. BioAPI is compliant with CBEFF, also standardized [14] as ISO/IEC 19785. This standard defines formatting of all biometric data in a structured way, called Biometric Information Record (BIR). BIRs are composed of a header, a body (called data block) and an optional security block (Fig. 2). Complex structures based on BIR can be built, by adding sub-headers to the data block.

All data exchange in the Evaluation System will be done using BIRs, as demanded by BioAPI. As it will be shown it is one of the requirements for the BSP development.

III. General Architecture of the Evaluation System

Following a top-down description, the solution developed is a complex Automatic Remote Evaluation System (ARES), where R&D community can access by submitting their algorithms encapsulated as a Biometric Service Provider (BSP). As shown in Fig. 3, those BSPs are loaded to ARES through a BSP Manager (BM) which also asks for further information, such as registered researcher's data, modality used, type of tests, databases to be used, etc. Once ARES finishes the evaluation, it will submit the results obtained through the Report Manager (RM) in a secure way to the developer submitting the information.

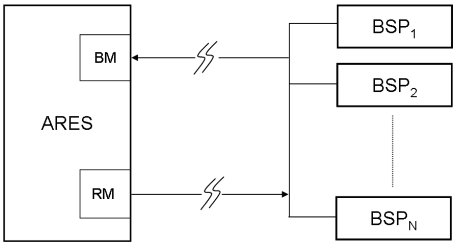

Fig. 3: Overview of ARES from the biometric developer

All communications are ciphered and authenticated using Public Key Infrastructure, based on the keys generated for the developer in

moment of registration. Such registration is done as a separate process and with manual supervision and acceptance, as to fulfill all legal issues related, such as Non-Disclosure Agreements (NDA).

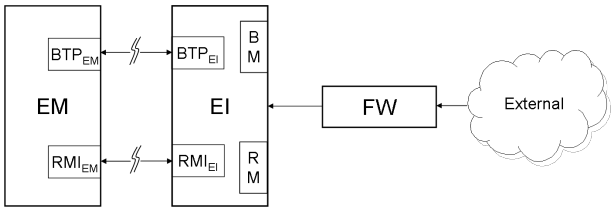

Fig. 4: ARES Modular Architecture

Internally ARES is based on 3 blocks, as shown in Fig. 4. First, a Firewall (FW) is used as to block as many external attacks as possible coming from the external world. Then a computer is used as an External Interface (EI), which can physically host also the FW. Such EI is in charge of processing petitions coming from the external world, hosting the registration application, and submitting reports generated by the system.

If a petition is fully accepted, then the EI transfers it to the Evaluation Module (EM) through a Biometric Transfer Protocol (BTP), requesting EM to enqueue it in the system. Through a Report Management Interface (RMI), EI receives the reports to be sent to the relevant developer. In order to increase security, EM is implemented in a separate computer, and BTP and RMI are proprietary, both in format and in physical communication layer. This also minimizes the effects that a potential attack of trying to access the information in EM, or trying to destroy the system.

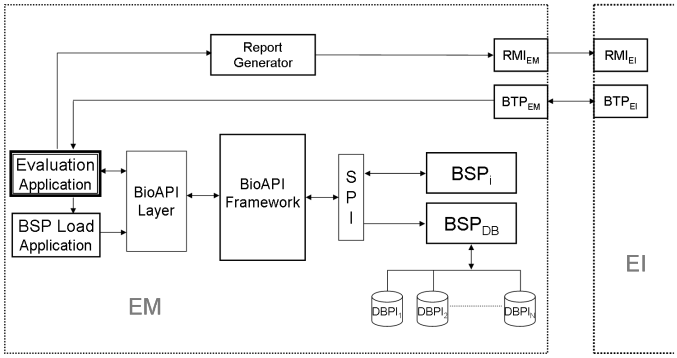

Fig. 5: Block Diagram of the Evaluation Module (EM) in ARES

Entering into details about how EM is implemented, Fig. 5 shows its block diagram. The most important block is the Evaluation Application, which controls the flow of all processes, as explained in the following section. The Evaluation Application is asked by BTPEM to perform an enqueued evaluation, sending the BSP and the parameters needed. It passes the BSP to the BSP Load Application, which will load not only the target BSP (with the developer algorithms), but also all Databases containing Personal Information (DBPI) needed for such evaluation. This will be realized through the BioAPI Interface Layer and using the BioAPI Framework.

Once all BSPs are loaded, the Evaluation Application will start its process, being fully compliant with BioAPI 2.0 (ISO/IEC 19784-1). Further details about the operational process within the Evaluation Application will be given in the next section.

IV. Operational Description

When designing the Evaluation Application, special focus has been placed in avoiding any data mining by the downloaded BSPs. Specially, no personal information from databases has to be extracted from the results reported. This can be solved by two means. First of all, extremely restricting the number and density of results given, which is non recommended due to the fact that such a decision will make the system not interesting for R&D community. The other solution is by not allowing the BSP to correlate the testing with data used, by randomizing the data used. Also, other techniques such as time spacing among tests requested by the same developers, and further protections are applied.

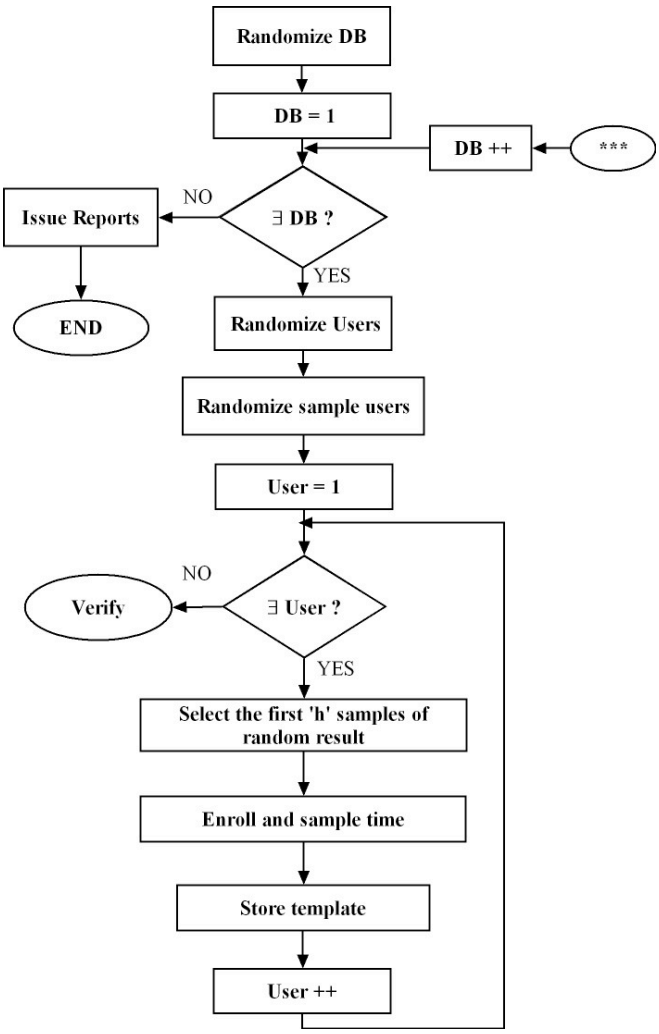

Fig. 6: Flowchart for the Evaluation Application

Therefore, the first step done by the Evaluation Application, as shown in Fig. 6, is to randomize, not only the users within one database, but also the databases requested for the testing. Once databases are randomized, for each database, users are also randomized, and each of those is enrolled to the test by generating their template using a set of h samples from the database (h is a parameter of the test, entered by the developer in the request form). Templates generated are stored in EM for later use in the Verify process. Once finished the enrolment process with all users, the Verify process is called. As with any other process related to the downloaded BSP, timing is always measured as to report it in the results.

The operational flowchart of the Verify process can be followed in Fig. 7. The initial task is to generate all comparison pairs needed for the test, removing from such a list all those used for template generation. Then, this list is randomized for applying tests in a non-deterministic order.

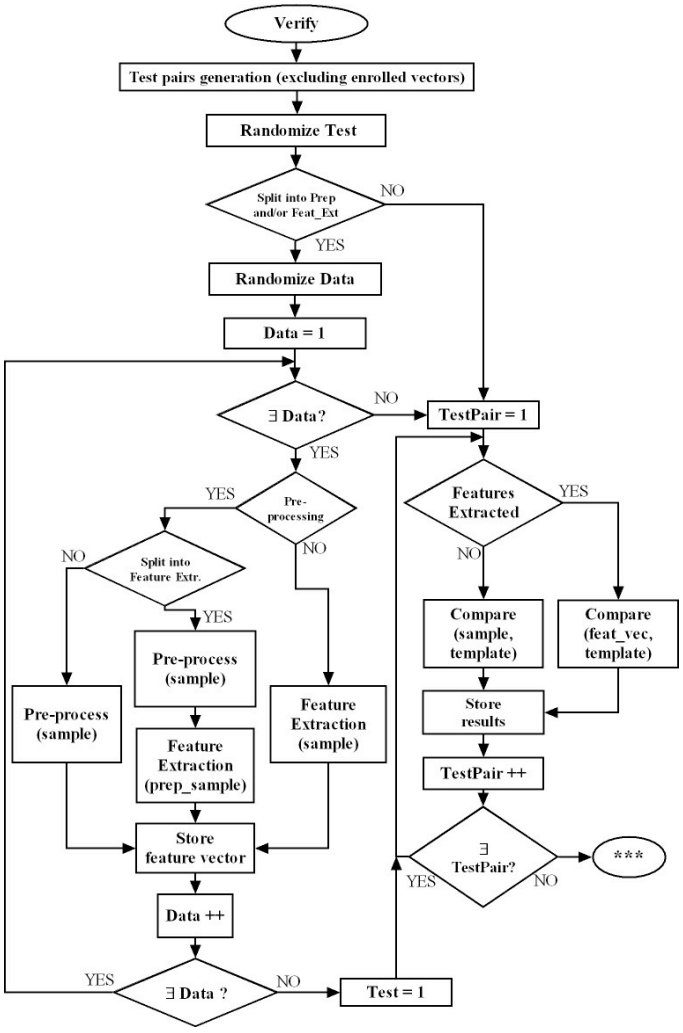

Fig. 7: Flowchart for the Verify process in EM

Since partial results may be needed, or because of trying to speed up the process, the BSP developer can supply functions to split Processing in Pre-processing and Feature Extraction, as well as providing different Comparison functions as needed. If this is allowed (which will be declared at the evaluation request form), then all samples in a database are pre-processed and/or their features are extracted, and stored outside the BSP. Once again, when performing such a task, samples are presented to the Pre-processing and Feature Extraction functions in a randomized way.

In case of no split of the verification process is declared, or when all samples have had their features extracted, comparisons start using all the test pairs. Depending on the algorithm, and how the BSP developer has arranged it, comparisons will be made directly with the samples and the previously enrolled templates, or with the features extracted against the templates.

Results given by the comparison function will be stored for further processing of the result reports. Although binary results are admitted, for the detailed results comparison scores are recommended. Another possibility is to provide binary results, but allowing the change of matching threshold in the comparison algorithm. This possibility is given for compliance with restricted environment development, but it obviously will have the disadvantage of requesting much more processing time.

Once all test pairs have been processed, control is returned to the Evaluation Application, as to proceed with the following database. As soon as all databases requested are processed, result reports will be generated as mentioned in section VI.

V. Target BSP Requirements

Those researchers, who would like to use ARES, will have to provide a BSP which shall cover a set of requirements. Those requirements, as shown in the following subsections, will deal with the format of the BSP uploaded, internal and exported functions and data formats accepted.

A. BSP Format and Exported Functions

The BSP has to be uploaded as a single DLL (Dynamic Link Library) file. The BSP has to have a unique valid identifier, and has to be able to export a valid schema (BioAPI_BSP_SHEMA), as to be able to be loaded in ARES.

If the BSP uses units, they have to be correctly declared and have to be included only in the Process and Matching categories.

The uploaded DLL file has to export a set of functions needed for the interaction with ARES. These functions compose the SPI interface, and are:

- BioSPI_BSPLoad
- BioSPI_BSPUnload
- BioSPI_BSPAttach

- BioSPI_BSPDetach
- BioSPI_GetBIRFromHandle
- BioSPI_Process
- BioSPI_VerifyMatch
- BioSPI_Free
- BioSPI_Cancel
- BioSPIRI_BSPGetSchema

B. BSP Processing Function

In a BSP uploaded, internal functions can be of two types: processing and verification. Processing functions have to be aware of how data is exchanged with ARES system, following BIR formats as described in section II. In order to develop such functions, implementation of BioSPI_Process is needed. To get samples from the Evaluation Module the function BioSPI_GetBIRFromHandle is needed, which, as expected, has also to be implemented.

As mentioned in section IV several implementations of the processing function may be used. BioSPI_Process should handle all the variants of the processing required, depending on its input and output parameters, and the design of the BSP:

- If ARES supplies the sample, and the BSP returns the feature vector, then feat_vec Preprocess(sample) will be used.
- If ARES supplies the sample, but the BSP returns a pre-processed sample, then the function used is prep_sample Preprocess(sample).
- Having ARES supplying a pre-processed sample, then feat_vec FeatureExtraction(prep_sample) is used.
- If not pre-processing is declared, then ARES can supply the sample, and obtain directly the feature vector from the BSP, by using the feat_vec FeatureExtraction(sample) function.

If the BSP considers only one of such variants (in such case only the last one is given), then BioSPI_Process can be called directly. However, when more than one possibility is implemented (e.g., when having the processing algorithm split in Pre-Processing and Feature Extraction), the authors recommend BioSPI_Process to analyze data as to know which possibility is to be called, and then forward such data to the relevant function, which can be called as mentioned in this paper.

C. BSP Verification and Matching Function

For verification the BSP has to implement BioSPI_VerifyMatch, where two BIRs are compared using a threshold determined by a claimed False Matching Rate (FMR). In BioAPI 2.0 there is also a high-level function, called BioSPI_Verify, which is not needed by ARES, because ARES is already supplying the samples, so no further capture is needed.

Analogously to the Process function, as mentioned in section IV, two different versions of BioSPI_VerifyMatch are considered:

- If verification is done directly using the sample provided by ARES, the function handled by BioSPI_VerifyMatch shall be Compare(sample, template).
- When comparisons are done using feature vectors, then the function to implement shall be Compare(feat_vec, sample).

Same considerations as with the Process function implementation are to be followed here.

D. Data Formats

Last but not least, BSP has to be aware that data is always used in ARES following CBEFF [14] and the relevant part of ISO/IEC 19794 series of standards related to biometric data formats. Both standards have to be followed at all times when interfacing EM in ARES. For intermediate results, such as feature vectors, and when no standard is available, such data has to be encapsulated in CBEFF format, indicating proprietary format. As such data will be removed from the system as soon as the evaluation ends, it is of no further importance the way such intermediate data is coded.

VI. Result Reports

Before ending the paper, some information about the result reports given will be presented in this section. As mentioned earlier in this paper, one of the important facts of the system is that it has to avoid any data mining attack from a fraudulent BSP, and at the same time providing the maximum amount of information needed by the developers. Therefore results returned are restricted to statistics calculated with the use of comparisons, thus giving no particular information about single comparisons. Also, some of the results are given only if the BSP functions provide feedback information such as sample rejection or quality scoring. Major results given, for each database used, include:

- Statistical data of the database
- Date and time of the initiation of the evaluation
- Date and time of the ending of the evaluation
- FTE: Failure to Enroll rate
- FTA: Failure to Acquire rate
- FNMR: False Non-Matching rate
- FMR: False Matching rate
- Enrolment time.
- Pre-processing time (when possible).
- Feature-extraction time (when possible).
- Comparison time (when possible).
- Verification process time.

- Intra-class distances.
- Intra-class timing.
- Inter-class distances.

All statistical data is given with its confidence levels. All timing and distances are given in mean, median, maximum, minimum, and standard deviation.

Reports are issued following the works being done in ISO/IEC JTC1/SC37 WG5, at the ISO/IEC 19795 multipart standard [15], which integrates the best practices for the evaluation of biometric systems, and its reporting.

VII. CONCLUSIONS

This paper has presented the work done by the authors in developing an Automatic and Remote Evaluation System (ARES), for helping R&D community with their developments, avoiding them to handle all legal issues related to data protection laws.

The implementation of such system, as well as its block diagram, has been presented. Also it has been described operationally through flowcharts. As interoperability and ease of use is demanded, authors have followed the most relevant standards existing nowadays, coming all of them from ISO/IEC JTC1/SC37 subcommittee on biometrics.

System is currently under legal assessment prior to acquire all personal information databases considered, and offering it to the R&D community. As a future working line, a security evaluation of the system, following Common Criteria is recommended, as well as improving tests and result reports. Furthermore, this work could be expanded to other areas where personal information is handled.

ACKNOWLEDGMENTS

This work has been partially funded by the PIBES Project, from the Spanish Ministry of Science and Education (TEC2006-12365), and also partially funded by the SEGUR@ Project.

REFERENCES

[1] European Commission, Justice and Home Affairs. Directive 95/46/EC. http://ec.europa.eu/justice_home/fsj/privacy/index_en.htm. European Union. 1995.

[2] NIST. "Special Databases and Software from the Image Group". http://www.itl.nist.gov/iad/894.03/databases/defs/dbases.html.

[3] NIST. "Iris Challenge Evaluation". http://iris.nist.gov/ICE/.

[4] NIST. "Face Recognition Grand Challenge". http://face.nist.gov/frgc/.

[5] K. Messer, J. Matas, J. Kittler, J. Lüttin, G. Maitre. "XM2VTSDB: The Extended M2VTS Database". Audio- and Video-based Biometric Person Authentication, AVBPA'99. 1999

[6] CASIA Iris Database. http://www.cbsr.ia.ac.cn/Databases.htm

[7] Univerzita Palackého. http://www.inf.upol.cz/iris/

[8] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, D. Escudero and Q.-I. Moro, "MCYT Baseline Corpus: A Bimodal Biometric Database", IEEE Proc.-Vis. Image Signal Process., Vol. 150, No. 6, December 2003.

[9] R. Cappelli, "SFinGe: an Approach to Synthetic Fingerprint Generation", in proceedings International Workshop on Biometric Technologies (BT2004), Calgary, Canada, pp.147-154, June 2004.

[10] Jinyu Zuo; Natalia A. Schmid; Xiaohan Chen; "On Generation and Analysis of Synthetic Iris Images". IEEE Transactions on Information Forensics and Security,. Volume 2, Issue 1, Page(s):77 – 90. March 2007 .

[11] Biometric System Laboratory, Pattern Recognition and Image Processing Laboratory, Biometric Test Center, Biometrics Research Lab – ATVS. "FVC2006: the Fourth International Fingerprint Verification Competition". http://bias.csr.unibo.it/fvc2006/.

[12] ISO/IEC JTC1/SC37. "ISO/IEC 19784-1, Information technology – Biometric application programming interface – Part 1: BioAPI specification". 2005

[13] BioAPI Consortium webpage: http://www.bioapi.org

[14] ISO/IEC JTC1/SC37. "ISO/IEC 19785, Information Technology - Common Biometric Exchange Formats Framework". 2005.

[15] ISO/IEC JTC1/SC37. "ISO/IEC 19795, Information technology – Biometric performance testing and reporting". 2005.