

Application-Specific Biometric Templates

Michael Braithwaite, Ulf Cahn von Seelen, James Cambier,
John Daugman, Randy Glass, Russ Moore, Ian Scott,
Iridian Technologies Inc.

Introduction

Biometric technologies that are capable of identifying individuals through one-to-many matching across large shared databases can provide convenient authentication services for many applications, including information security, physical access, financial services, etc. without requiring costly and time-consuming re-enrollment for each application. But the potential for shared access and multiple uses of biometric databases raises serious concerns with respect to personal privacy because biometric templates are considered by some to constitute personal information that could be used for unauthorized purposes. The standardization of template formats, intended to promote deployment by enabling sharing of costly enrollments, has in addition created a security vulnerability. An enrollment or recognition template created for one purpose could be misappropriated and used for fraudulent purposes. And unlike a PIN or password, a biometric template cannot be changed, recovered, or reissued if it is compromised [1]. By their nature, biometric entities are stable over time, otherwise their utility would be quite limited.

One potential means of protecting stored templates is encryption, but because the matching algorithms used to match templates and thereby authenticate an individual identity cannot, in general, operate on such encrypted templates, the templates must be decrypted prior to matching. Thus the decrypted templates are inevitably exposed to potential hacker attacks when matching is being performed. The management and protection of private keys also presents challenges that are well-documented. Furthermore, cryptographic algorithms can be computationally expensive and limit the capacity of large-scale biometric systems to provide responsive authentication services.

One proposed solution to the problem of compromised templates is the introduction of predefined distortions of raw biometric data or extracted features [2]. When applied to image-based biometrics like fingerprints or facial recognition, this technique has the potential for enabling re-issuance of templates. Because the transformations are intended to be nonreversible, however, the possibility of converting a database from one specialized format to another may be limited. In addition, it is necessary at least in some cases to reverse the transformation prior to matching; this exposes the original biometric data to hacking during the matching process and may represent a significant vulnerability.

The technique described here is based on the definition of unique, application- (or even transaction-) specific formats for biometric templates that prevent the unauthorized exchange of templates across multiple applications, yet provide a mechanism for authorized transfer across applications. In addition they support the re-issuance of compromised templates without re-enrollment. Finally, the template matching operations are invariant across the transformations, so there is no need to return templates to a vulnerable “nontransformed” state in order to perform authentication.

Biometric Template Transformations

We describe here a means for transforming a biometric template so that it assumes a new format that is unique to a particular application. Such a transformed template cannot be successfully matched to a second template extracted from the same biologic entity unless the second template is transformed so that its format is identical to that of the first template. Thus a template generated in a format corresponding to a particular application A could not be misappropriated and used to authenticate a user for application B because the enrollment database for application B would have a different format than those enrolled for application A.

Consider biometric templates T_1 and T_2 derived from the same biologic entity (hand, finger, eye, etc.) such that an appropriate matching function $M(T_1, T_2)$ has a value

$$M(T_1, T_2) = 1$$

if the templates are judged to match (i.e. to have come from the same biologic entity) and

$$M(T_1, T_2) = 0$$

if the templates are judged to not match. Assume first that templates T_1 and T_2 are generated in exactly the same way with the same format so that if they do indeed come from the same biologic entity, $M(T_1, T_2)$ will have a value of 1.

Now we apply a transformation F_A to the “root” templates T_1 and T_2 so that the transformed templates $F_A(T_1)$ and $F_A(T_2)$ have a unique format specific to a particular use or application A. We desire that the transformation F_A have the property that the matching process is invariant under the transformation, that is,

$$M(F_A(T_1), F_A(T_2)) = M(T_1, T_2)$$

This invariance is important because it means that matching can be performed on the transformed templates, making it unnecessary to reverse the transformation, recreating and exposing the root templates T_1, T_2 prior to or during the matching process.

Transformation Properties and Benefits

If two different transformations F_A and F_B are defined for applications A and B we need

$$M(F_A(T_1), F_B(T_2)) = 0$$

for T_1, T_2 from the same biological entity while in that case

$$M(F_A(T_1), F_A(T_2)) = 1 \text{ and}$$

$$M(F_B(T_1), F_B(T_2)) = 1$$

This property assures that a template generated for one application A cannot be used for another application B.

Figure 1 illustrates the process of enrollment, by which a database of enrolled templates having a format defined by F_A is created. Biometric data from the user is processed to create a root enrollment template having a standard format; this root template is then transformed using transformation $F_A(T_1)$ so its format is now specific to Application A. Note that it is possible, with suitable software design, to create the transformed template directly without ever generating the root template, by incorporating the transformation process into the template generation process. This avoids possible exposure of the root template. The enrollment template, and secondary identifying information, if desired, are stored in the enrollment database for Application A. Note also that this enrollment process is equally applicable to both verification and identification systems.

The recognition process associated with Application A is shown in Figure 2. A root template is created and transformed to conform to the prescribed format for Application A and, as before, it is possible to do so without explicitly creating the root template. The match function $M(F_A(T_1), F_A(T_2))$ compares the just-created template with one or more templates from the Application A database, depending on whether the system is performing verification or identification. Note that although the example shows generation of only a single user template, it is equally appropriate to generate multiple user templates, representing multiple samples of the same biometric entity, or samples of multiple biometric entities (i.e. multiple eyes or fingers) and match each against the Application A database, thus accounting for variability in the template generation process that may cause some templates to be falsely rejected.

If a template from Application A were used to attempt to authenticate to a database created for Application B, the match function, comparing templates with different formats, would nearly always return a zero, indicating no match. The probability of such a match returning a value of one will be no greater than the likelihood of two randomly selected templates matching, i.e. the single-match false accept probability of the biometric technology. In the case of exceptionally strong biometric technologies like iris recognition [3-5] this probability is extremely small. This is true even if the two templates T_1 and T_2 are from the same biologic entity and indeed even if T_1 and T_2 are identical.

The format transformations anticipated have the further property that they can be processed to create new templates. Hence if we have template $F_A(T_1)$ we can define transformation $F_{A,B}$ such that

$$F_B(T_1) = F_{A,B}(F_A(T_1))$$

$$\text{i.e. } F_{A,B} = F_B F_A^{-1}$$

where F_B is the format created for Application B. If a user has created an enrolled template for application A, he or she can request the custodian of database A to make his or her enrolled template available to the Application B database after application of transformation $F_{A,B}$ to change its format. Responsibility for definition and application of transformation $F_{A,B}$ can rest in a trusted format authority that maintains a registry of formats and defines and applies the transformations required to convert templates from one format to another, subject to user authorization. The advantages of this process are twofold: (1) no application transformation is ever exposed to any other application, and yet (2) users can use their existing enrollments for new applications without incurring the cost and inconvenience of re-enrolling their biometric for each new application. Such transformations would be performed only if specifically requested and authorized by the user who produced the original template. Indeed the biometric itself can be used to authorize the transfer of the enrollment template.

Another benefit of the transformation technique is that if a database custodian suspects or determines that its biometric data has been compromised, or its format has been discovered, it can request that the Template Authority define a new transformation for its entire database, changing its format and rendering the stolen templates completely useless. Functionally, this is equivalent to changing a password if it is determined that the password has been compromised. This process is illustrated in Figure 3. Upon receiving a request from Application A for a new format, the Template Authority creates a transformation F_C that will serve as the new transformation for Application A. Using the archived transformation for Application A, F_A , the Authority generates its inverse and processes it with F_C to form $F_C F_A^{-1}$. This latter transformation is the Conversion Transformation, which will be applied to the entire Application A database to convert its enrollment templates to the new C format. At the same time, the user transformations must be updated to reflect the change in format from A to C. Managing these updates can be automated by incorporating a format revision code into the user's submitted template and advising the user of the need to download a new transformation when an obsolete format is detected.

Figure 4 illustrates a client-server variation on the simple authentication process shown in Figure 2. Here we assume that the user has previously enrolled for application "A" as before and the database for A contains the enrollment template. When the user wishes to be authenticated he generates a request to the server for a unique transformation "seed" number or key. The server generates a random seed denoted in Figure 4 as "X". Recall that the enrolled templates have been previously transformed by F_A to conform to the format defined for application "A". The server transmits the seed X to the client and at the same time computes the transformation

$$F_{X,A} = F_A F_X^{-1}$$

and saves it in temporary storage. It then deletes X, F_X , and F_X^{-1} . The client, upon receiving X, uses it to generate its own copy of F_X . It then captures an image and generates a biometric template using F_X to transform the root template T_1 to the format prescribed by X. This template, which we designate $F_X(T_1)$ is digitally signed and encrypted if desired, and then transmitted to the server. The server, after decrypting

the template and verifying its integrity using digital signature techniques, uses its temporarily stored transformation $F_{X,A}$ to convert the client's template to a format compatible with database A:

$$\begin{aligned} F_A(T_1) &= F_{X,A}(F_X(T_1)) \\ &= F_A F_X^{-1}(F_X(T_1)) \end{aligned}$$

The client's template has been generated and transmitted to the server in a unique format valid for only this single transaction. Only the server has the information needed to render $F_X(T_1)$ compatible with its enrollment database.

Yet another form of client-server authentication is shown in Figures 5 and 6. Again the user has previously enrolled in the database for application "A". But in this case before the enrollment is performed, the client application generates a random seed number and computes its own unique "A" transformation as shown in Figure 5. This transformation is applied to the enrollment template before sending it to the server. It is also stored on a smart card or other portable media that the user keeps in his possession. Note that now the user may perform enrollments for a number of applications, each time saving the appropriate transformation in portable storage. Plus, each template in the enrolled database will have its own unique format, known only to the user. The user thus has complete control over the use of his or her own biometric data because the unique format of the biometric template is defined by the transformation stored on the portable media. When authentication for application "A" is required, the user captures an image with the appropriate biometric device, generates a root template, and inserts the portable media for the "A" application into an appropriate reader. This process is illustrated in Figure 6. The client application reads in the transformation, applies it to the root template, and sends the transformed template to the server. As before, the transformed template may be encrypted and digitally signed prior to sending it to the server.

Transformations

The transformation technique described here relies on biometric templates that are composed of an array $[t_1 t_2 t_3 \dots t_n]$ of independent data entities t_i , which may be isolated binary bits or groups of bits. Templates of this type are used in iris recognition [3-5] and have been proposed for fingerprint systems [6]. A compatible matching function is one that judges the similarity between two templates by examining corresponding independent data entities. One such function is the Hamming Distance $HD(T_1, T_2)$ which examines every pair of corresponding bits in templates T_1 and T_2 and counts the proportion of bits that differ between the two templates. The HD value is then compared to a threshold to generate a binary match result. The HD concept can be generalized to larger data entities, counting the number of corresponding entities that are not identical. For example, bits might be examined in groups of 2 bits, in which one bit represents a data value and the second bit a control bit indicating the validity of the data bit. In this case, the two data bits are compared and used in the HD calculation only if both control bits have a value confirming the validity of the data bits.

A suitable transformation F used for such biometric templates must have three properties:

1. F must not alter the length of the template
2. F must not change the value of the control bits, if used
3. F must not alter the number of matching (or mismatching) data bit pairs

Conditions (1) and (2) are easily fulfilled, but (3) is more difficult to meet. F must transform each valid data bit independently of the value of any other data bit. One preferred transformation simply alters the position of some or all data bits. This is simple permutation, and if a template consists of n independent entities, there are $n!$ possible transformations. For example, if the data entities are 8-bit bytes, and there are 256 data bytes in each template, the number of possible permutations is $256!$ which is estimated using Stirling's approximation to be 8.6×10^{506} . If the data entities were single bits instead of bytes, the number of permutations would be $2048!$ which is estimated as 10^{5894} . It is preferable to use only transformations that alter the position of every data entity, preventing the possibility of false matches. Such permutations

are termed “derangements” and their number is somewhat less than the total number of possible permutations. The number of possible derangements of 256 data elements, for example, is about 6.2×10^{506} . All such permutations possess readily-computed inverses.

A second preferred form of transformation, applicable only to single-bit data entities, is based on the logical exclusive-or (XOR) function. In this transformation single bit values are XORed with a predefined mask function. If T_i is the i^{th} data bit of template T and M_i is the i^{th} mask bit then the i^{th} transformed template bit is

$$F_i(T) = T_i \text{ XOR } M_i$$

The XOR function changes the value of any bit for which the corresponding mask bit is a 1. If the template has 2048 data bits, for example, the number of possible masks is $2^{2048} = 3.2 \times 10^{616}$. Preferably the mask contains 1's in at least half its positions to avoid ineffective transformations that do not significantly affect the template. The number of such transformations is 1.6×10^{616} . The XOR function serves as its own inverse.

It is also possible to combine transformations of different types. Thus a permutation could be followed by a logical XOR transformation, further enhancing the security of the templates and increasing the number of possible forms of transformation.

The extremely high number of possible, unique transformations of the biometric template makes the scheme highly effective against brute force attacks (trial and error of all possibilities).

Summary

We have described techniques for transforming biometric templates that preserve the accuracy and flexibility of strong authentication technologies, assure the privacy of biometric databases, permit the sharing and re-use of enrolled templates if authorized by their owner, and enable the reissuance of biometric templates that have been compromised. These transformations have the potential for resolving major issues of privacy and security in biometrics, particular for those biometric technologies that are able to offer the benefits of identification-based authentication with large databases.

References

1. Schneier, B. *Secrets & Lies: Digital Security in a Networked World*, New York, John Wiley & Sons, pp 141-145, 2000.
2. Ratha, N. and Connell, J., “Cancelable Biometrics”, presented at Biometric Consortium 2000 Conference, Sept. 13-14, 2000.
3. Daugman, J. “High confidence visual recognition of persons by a test of statistical independence”, *IEEE Trans Pattern Analysis and Machine Intelligence*, 15(11):1148-1161, 1993.
4. Daugman, J. “Recognizing persons by their iris patterns”, in *Biometrics: Personal Identification in Networked Society*, A. Jain, R. Bolle, S. Pankanti, eds., Amsterdam, Kluwer, pp 103-121, 1998.
5. Flom, Leonard and Safir, Arin, “Iris Recognition System”, U.S. Patent No. 4,641,349.
6. A. K. Jain, S. Prabhakar, L. Hong and S. Pankanti, “Filterbank-Based Fingerprint Matching”, *IEEE Transactions on Image Processing*, 9(5):846-859, 2000

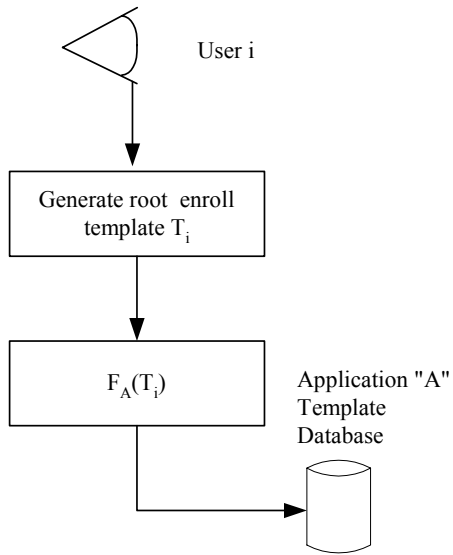
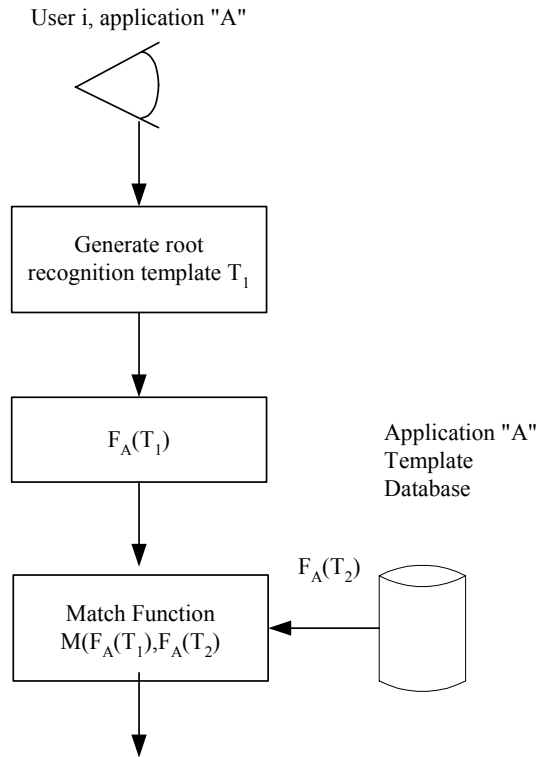


Figure 1 Enrollment Function



Result = 1 if T_1, T_2 from same biological entity else
 Result = 0

Figure 2 Match Function

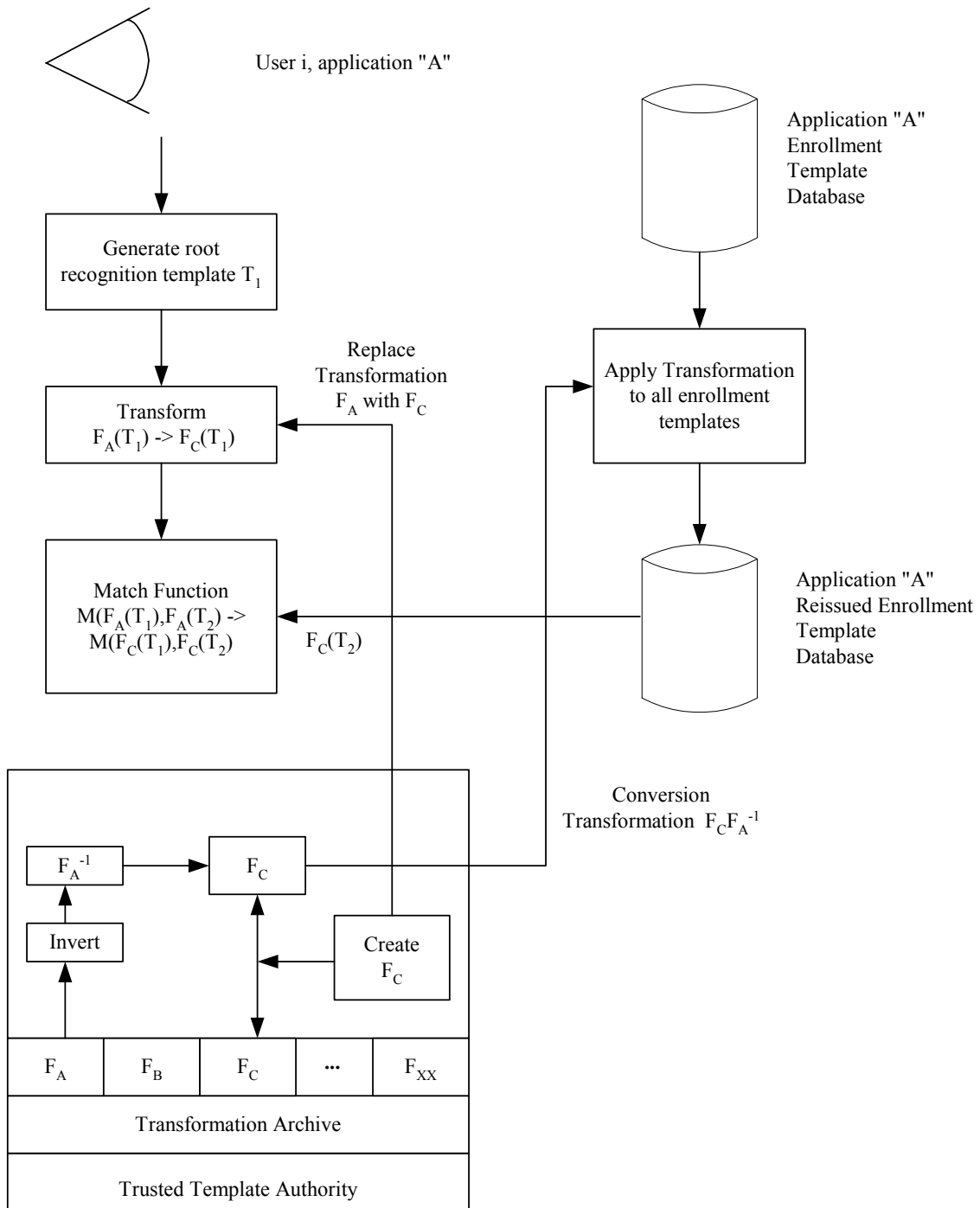


Figure 3 Template Database Reissue

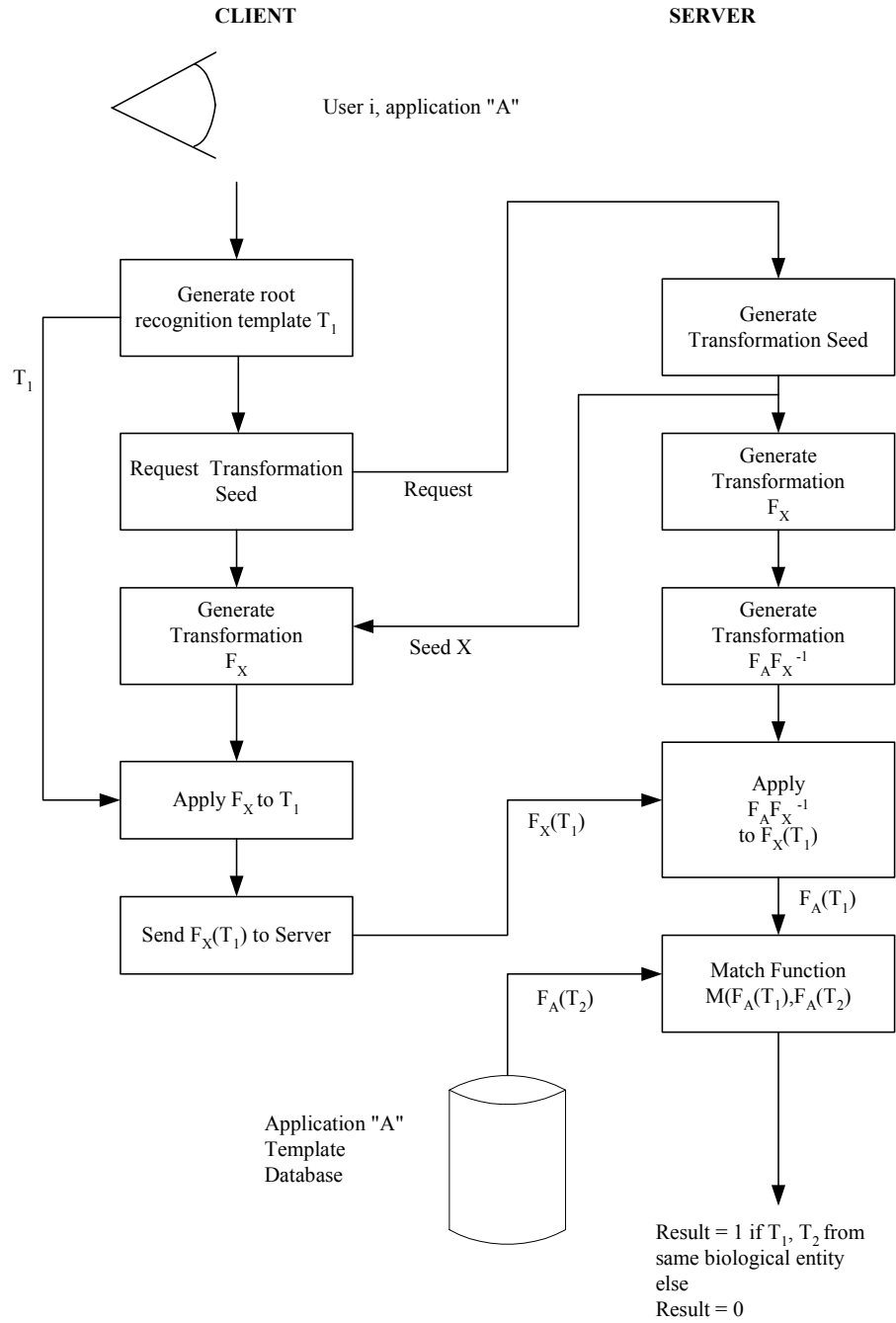


Figure 4 Client-Server Authentication

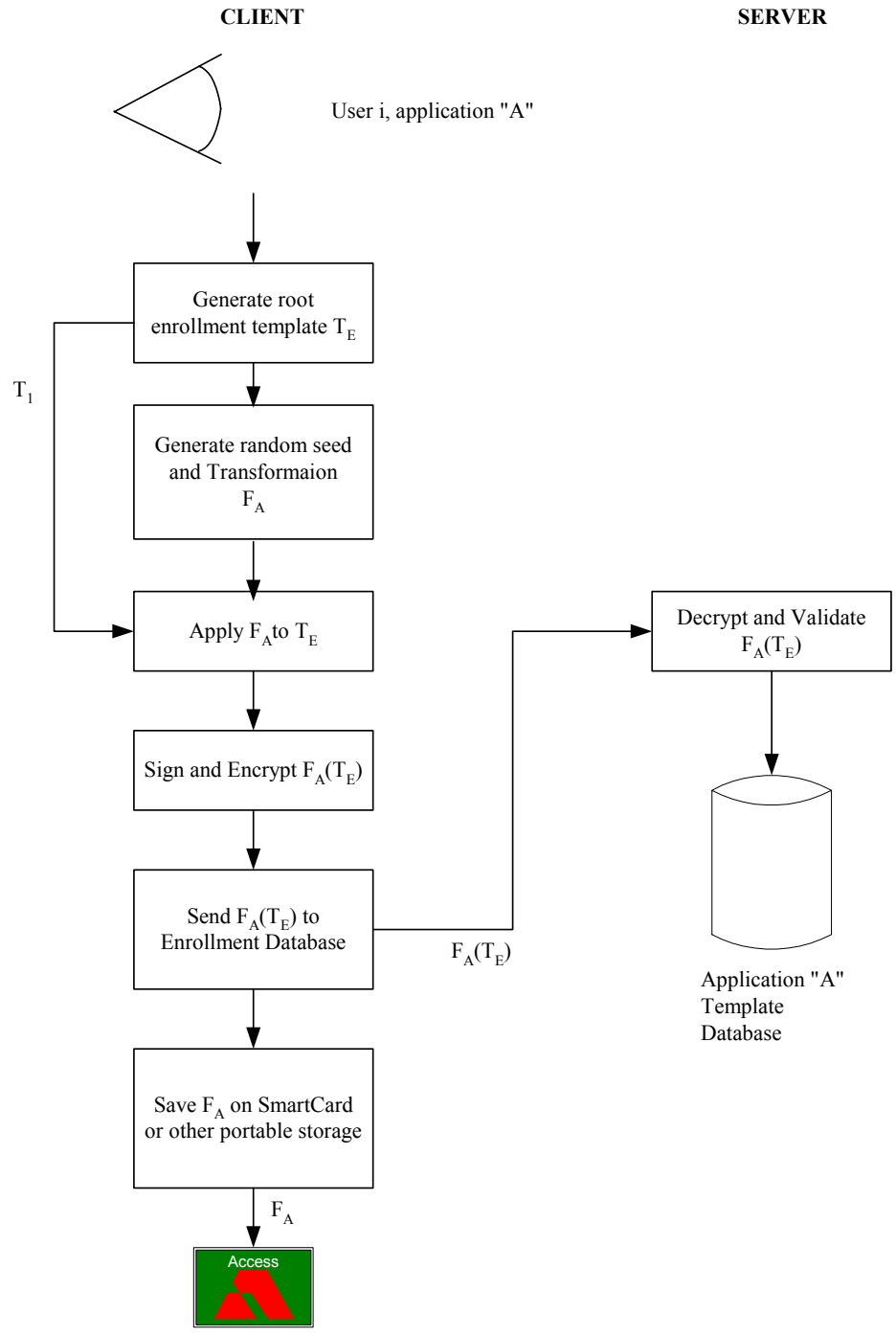


Figure 5 Client-Server Enrollment

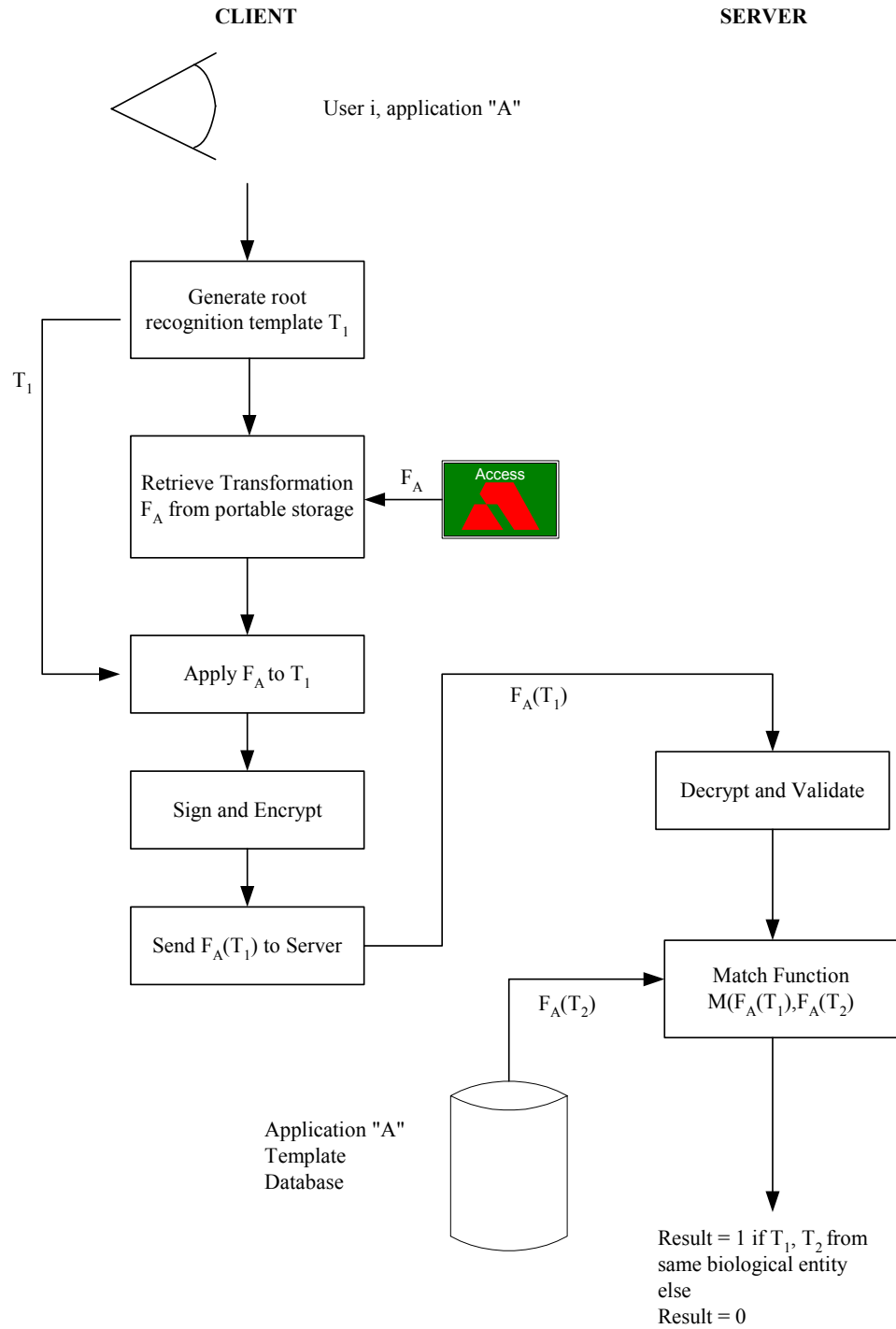


Figure 6 Client-Server Authentication