Feasibility of generating biometric encryption keys

S. Hoque, M. Fairhurst, G. Howells and F. Deravi

A scheme to generate encryption keys from biometric samples is introduced. The idea is based on partitioning feature space into subspaces and partitioning these into cells. Each cell subspace contributes to the overall key generated. Experimental analysis shows encouraging prospects.

Introduction: Security is a major concern in modern society, especially given the utilisation of digital techniques in the creation, editing and distribution of sensitive data. Digital data can easily be copied and multiplied without any information loss, but the ubiquity of the Internet makes it very difficult to control and trace such intrusions by unauthorised people. Furthermore, e-commerce applications must ensure highly secure mechanisms for user identification and controlled access to transmitted sensitive data. Primary security concerns are mainly confidentiality, data integrity, data origin authenticity, agent authenticity, non-repudiation, etc. Many currently available cryptographic techniques can address these issues and PIN, password, smartcards, etc., along with public key infrastructure (PKI) have been used widely for this purpose. Unfortunately, the security of a robust encryption mechanism lies in the ability to keep its cipher key(s) secret, while typical human behaviour patterns (writing down keys and choosing trivial or obvious passwords) often makes security extremely vulnerable to compromise. Also, such techniques cannot identify a person beyond all doubt and there is then considerable scope for fraudulent usage. Hence, further identity authentication measures (such as biometric verification) are often required. Live biometric samples can reliably verify not only user identity but also their physical presence at a remote station.

It is useful therefore to investigate whether the present two-tier approach can be merged into a one-step process in such a way that encryption key(s) are extracted directly from the biometric samples. Having a biometric-based key offers a number of distinct advantages, including the fact that this removes the need for a user to remember their keys and, as a biometric is an inherent feature of a human user, it cannot be transferred to a third party. The difficulty lies in the fact that all encryption algorithms expect the key entered to be exactly the same on every use, and this is clearly not the case for a typical biometric. In addition, ageing, illness, environmental factors, etc., have a bearing on the quality and variability of captured biometric data.

State of the art: To overcome the variability in the biometrics to generate the encryption key, a number of approaches are possible. The simplest is to hide the keys in the biometric templates as payloads. A successful biometric identity verification transparently releases this key. Many commercial biometric programs support this approach, and Soutar *et al.* [1] also introduced a similar scheme. A second approach would be to generate the key from this template using a suitable one-way function. Bodo [2] first proposed that data derived from the template be used directly as a cryptographic key and Janbandhu *et al.* [3] and Daugman [4] supported this notion. As sample variability has no direct bearing on these templates, the same key can be generated at all times, but a major drawback of the approach is that, if a user needs to change his template (e.g. due to ageing), the previous key may never be regenerated. Uludag *et al.* [5] proposed a multiphase protocol where a key is developed during the initial handshaking with the server. The key is a function of the data as well as the server and user identity (biometric). This scheme assumes the presence of a trusted handler that prevents decryption if biometric verification failed. Vector quantisation (VQ) can be used to counter the variations in a user's biometric samples. The feature space is partitioned into a number of cell spaces, each cell space denoted by mean vectors. Live samples are compared with these vectors. The nearest vector in the codebook determines the user identity and can then be used to extract the encryption key. Yamazaki *et al.* [6] described such a method to generate a biometric-based key.

Proposed approach: The method introduced here is a modification of the VQ approach. Here the codebook is replaced by a series of partitions induced in the feature subspaces, each created by a subset of feature dimensions and the partitions define a number of cells in these subspaces. Each cell is tagged with a key component (usually its identity). When a live sample is available, it is checked against these partitions to ascertain its membership of a cell. Each feature subspace (denoted by its own set of partitions) is treated independently and contributes its share of the encryption key. As there are many subspaces, by concatenating these key segments a complete key can be obtained. In this proposition, users do not need to declare individual identities to have access to a secured file. The capability to provide a biometric sample that can decipher the file is an acceptable proof of identity. On the other hand, the partitions are created based on feature-point distribution in the subspace (rather than user identities). Therefore, multiple users may share the same cell space, and their biometrics will lead to the same encryption key. Such unintended impersonation, where an individual tries to access a document secured by his cellmate, is found to be very unlikely.

Complexity analysis: Assume the feature space to be N-dimensional with K the desired number of possible keys, implying that the feature space must be split into K cells. We may create L subspaces each of n_i dimensions ($1 \le n_i \le N$, $1 \le i \le L$) where $L = \sum_i \{{}^N C_{ni}\}$. If each is partitioned into k_i cells, the key size will be $\log_2 K = \sum \log_2 k_i$ bits. For $N = 5$, $k_i = 8$, a key size of 93 bits is possible. To ascertain the cell membership, the feature vector needs to be compared with a number of partition descriptors. Hence the complexity of the key generation process is $C(N) = L\ k_i\ T(i)$. Here, $T(i)$ is the complexity of comparison in an i-dimensional subspace. It is possible to use only a subset of the available subspaces (e.g., using only the one-dimensional subspaces). Since $K = f(N,k_i)$, to maintain the same key size, a larger feature vector and/or more partitions per feature space would be necessary.

Experimental evaluation: To establish the feasibility of the proposed technique, a number of empirical investigations were designed. Synthetic features were used to represent an idealistic scenario, with the objective of establishing the desirable characteristics of the extracted features. For further simplicity, only one-dimensional feature subspaces were initially adopted. The experimental database consisted of feature vectors for 400 users each giving ten samples, uniformly distributed in the feature space (i.e. the inter-class distribution is uniform) whereas intra-class distribution is assumed to be Gaussian. To determine the impact of intra-user variability, the standard deviations of individual users (σ_u), although generated at random, were subjected to a predetermined maximum. Five samples from each user (chosen at random) were used for training and the remainder for testing. To know the probabilities of accidental impersonation (or identity confusion), an imposter database is also generated. A user is chosen at random to act as an impersonator with an assumed identity also chosen at random. The imposter database contained 5000 such samples. These database samples were used in the determination of system false acceptance error rates (FAR).

Table 1: Mean FRR (%) from partitioning 1D feature vectors

Max(σ_u)	k_i			
	2	4	8	16
≤ 0.005	0.12	0.39	1.01	2.31
≤ 0.01	0.39	1.15	2.76	6.28
≤ 0.02	0.98	2.32	5.33	12.59
≤ 0.05	2.27	6.56	17.48	31.51
≤ 0.10	4.56	13.96	32.08	48.51

We initially investigated how useful these partitions were and their relationship with the feature characteristics. Table 1 demonstrates the failure of a genuine user to be positioned in the cell space assigned to them (FRR). This cell assignment was determined during training as the subspace cell containing most of their training samples. It is clear from Table 1 that on average FRRs more than doubled as the number of partitions was doubled. At the same time, FRRs also deteriorated by more than a factor of 2 when σ_u was doubled. Hence, when intra-user variability is high, it is better to use fewer partitions, and variability beyond a certain limit renders that feature subspace useless because these errors accumulate when key segments from other subspaces are concatenated. Ideally, we should aim for $\sigma_u \ll 1/k_i$ in order to have an

acceptable FRR at 1D subspaces. Table 2 presents the mean FAR, the probability of identity swap. One interesting observation is that σ_u has no impact on these errors. The number of partitions has an effect to the extent that these error rates are identical to that of the success rate of random guessing.

Table 2: Mean FAR (%) from partitioning 1D feature vectors

Max(σ_u)	k_i			
	2	4	8	16
≤ 0.005	50.04	25.18	13.33	6.42
≤ 0.01	50.48	25.55	12.53	6.33
≤ 0.02	49.62	25.53	12.58	6.81
≤ 0.05	50.34	25.70	12.56	6.40
≤ 0.10	50.20	26.17	13.36	6.85

It is always desirable to have low FRR and FAR. This creates a dilemma because they contra-indicate with respect to the selection of the number of subspace partitions. Appropriate optimisation criteria therefore need to be developed. Hence, we investigated the effect of concatenation of multiple subspace identities on the FRR and FAR, as shown in Table 3. One interesting observation is that, irrespective of user variability, number of partitions or overall feature dimensionality, FAR dropped to zero. Since concatenation of the results from subspaces is an integral part of the proposed system, we can expect 0% FAR. Table 3 also reveals that similar key spaces can be generated using different settings, resulting in significantly different FRR. For example, a 32 bit key can be generated either from $\{N=32, k_i=2\}$ or $\{N=16, k_i=4\}$ or $\{N=8, k_i=16\}$, resulting in FRRs of 12.6, 16.5 and 41.1, respectively (for $\sigma_u=0.01$). The same pattern is observed for other σ_u, emphasising the importance of large N. Since most robust applications require a key much larger than 32 bit, Table 4 shows what is achievable when, for example, a 128 bit key is required.

Table 3: Mean FRR and FAR (%) after concatenation of 1D partitions

Max(σ_u)	K_i	FRR max			FAR
		$N=8$	$N=16$	$N=32$	
0.005	2	0.9	2.2	5.6	0
	4	3.1	5.3	13.7	0
	8	7.8	16.8	27.1	0
	16	17.9	30.8	51.0	0
0.01	2	3.2	6.3	12.6	0
	4	8.8	16.5	27.3	0
	8	20.0	35.8	54.0	0
	16	41.1	63.7	83.1	0

Table 4: Mean FRR (%) for 128 bit key

Configuration	FRR			
	$\sigma_u=0.001$	$\sigma_u=0.005$	$\sigma_u=0.01$	$\sigma_u=0.02$
$\{N=128, k_i=2\}$	1.3	19.5	37.5	65.8
$\{N=64, k_i=4\}$	3.3	25.0	50.6	76.1

It is clear that useful encryption keys may be generated from biometrics depending on the extracted feature variability. For example, extreme sparseness in populating IrisCode space [4] indicates availability of suitable features in the biometric domain. Errors can be further reduced by allowing users to make multiple attempts in case of failure or considering neighbouring key spaces as possible candidates. It is also possible to use multimodal biometrics and to concatenate password/PIN with the biometric key to substantiate and/or further enhance the key space.

Conclusion: A possible scheme to extract encryption keys from biometric samples is presented. Experimental results highlight the principal desirable characteristics in the biometric features and point to an optimal system configuration.

21 October 2004
Electronics Letters online no: 20057524
doi: 10.1049/el:20057524

S. Hoque, M. Fairhurst, G. Howells and F. Deravi (*Department of Electronics, University of Kent, Canterbury, Kent CT2 7NT, United Kingdom*)

References

1 Soutar, C., *et al.*: 'Biometric encryption' *in* Nichols, R.K. (Ed.): 'ICSA guide to cryptography' (McGraw-Hill, 1999)
2 Bodo, A.: 'Method for producing a digital signature with aid of a biometric feature', 1994 (German Patent DE 4243908A1)
3 Janbandhu, P.K. *et al.*: 'Novel biometric digital signatures for internet based applications', *Inf. Manage. Comput. Secur.*, 2001, **9**, (5), pp. 205–212
4 Daugman, J.: 'Biometric decision landscapes', Technical Report TR482, 2000 (University of Cambridge Computer Laboratory, Cambridge, UK)
5 Uludag, U. *et al.*: 'Multimedia content protection via biometrics-based encryption'. Proc. ICME2003, Maryland, USA, 2003
6 Yamazaki, Y. *et al.*: 'A secure communication system using biometric identity verification', *IEICE Trans. Inf. Syst.*, 2001, **E84-D**, (7), pp. 879–884