# Secure Continuous Biometric-Enhanced Authentication

Andrew J. Klosterman Gregory R. Ganger

May 2000

CMU-CS-00-134

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

We thank the members and companies of the Parallel Data Consortium (including CLARiiON, EMC, HP, Hitachi, Infineon, Intel, LSI Logic, MTI, Novell, PANASAS, Procom, Quantum, Seagate, Sun, Veritas, and 3Com) for their interest, insights, and support. This work was also supported in part by the General Motors Corporation through the GM/CMU Satellite Research Laboratory. Further support came from the Pennsylvania Infrastructure Technology Alliance, a joint program of Carnegie Mellon and Lehigh University funded under the Commonwealth of Pennsylvania's Department of Community and Economic Development. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the respective centers, companies or universities.

**Keywords:** Computer Security, Authentication, Biometric, Eigenface.

**Abstract**

Biometrics have the potential to solidify person-authentication by examining "unforgeable" features of individuals. This paper explores issues involved with effective integration of biometric-enhanced authentication into computer systems and design options for addressing them. Because biometrics are not secrets, systems must not use them like passwords; otherwise, biometric-based authentication will reduce security rather than increase it. A novel biometric-enhanced authentication system, based on a trusted camera that continuously uses face recognition to verify identity, is described and evaluated in the context of Linux. With cryptographically-signed messages and continuous authentication, the difficulty of bypassing desktop authentication can be significantly increased.

# 1 Introduction

Over the years, many elegant authentication schemes and powerful access control mechanisms have been developed. With these, a system can decide whether a given "principal" has the necessary privileges to perform a given action. When humans are involved, the first step becomes associating a person requesting access with an on-line personality that becomes the principal in subsequent activities. Unfortunately, most systems continue to use easy-to-compromise person-authentication mechanisms. Specifically, most current desktop person-authentication relies upon passwords, which have repeatedly been shown to be susceptible to theft, guessing, and sharing [30].

Many promote biometrics as the solution to person-authentication problems [3, 8, 18]. Biometrics are values that encapsulate biological features (e.g., a fingerprint or iris pattern), and ideal biometrics differentiate people as precisely as the original physical features. On the surface, use of such unique-per-person features would seem to allow exact person-authentication, solving this long-standing security problem. However, there are a number of technical and social issues that must be addressed in using biometric-based authentication. While they offer intriguing person-authentication benefits, poor integration of biometrics into systems can yield reduced privacy and increased user irritation with little increase in security. Unfortunately, many current systems misuse biometrics and therefore provide limited security benefits.

This paper explores the use of face verification to enhance the person-authentication of desktop computer systems. Using a hybrid Linux and Windows NT based prototype, we explore the performance and usability of the state-of-the-art. In describing our system, we discuss issues related to secure and effective integration of biometric-based authentication. Most importantly, biometrics are not like passwords and cannot be used in the same way. In addition, our system uses the non-intrusive nature of video-based authentication to continuously verify the identity of its users. This provides much more robust authenticity than current log-in/log-out session approaches.

A major challenge with biometrics is the fact that they cannot be protected (practically). They are not secrets, like passwords, and they are not physical items, like tokens. As a result, they cannot be used by the system in the same way as these others. Simply presenting a correct biometric is not evidence of identity, because anyone with access to a person can obtain most non-intrusive biometrics from them. For example, facial images can be obtained from camera snapshots or pictures downloaded over the web. Similarly, fingerprints can often be lifted directly from stolen laptops. Unless precautions are taken, using these biometrics does not require fooling the biometric sensor — one can simply bypass the sensor entirely and provide pre-captured biometric data directly. Thus, a key to secure use of biometrics is to make specific biometric sensors be trusted (and trustworthy) components within systems and to verify authentication claims back to these original points.

The remainder of this paper is organized as follows. Section 2 describes person-authentication schemes in general and how biometric-based authentication differs from traditional schemes. Section 3 discusses design issues for secure and continuous biometric-based authentication. Section 4 describes the implementation of our system and its integration into Linux. Section 5 evaluates the accuracy, usability, and computational overhead of our prototype. Section 6 discusses a number of additional issues related to biometric-based authentication. Section 7 discusses related work. Section 8 summarizes this paper's contributions.

## 2 Using Biometrics for Authentication

The process of authenticating people as computer system personas has always relied on verifying an association between the persona and one or more of the following:

1. Something a user *knows* (e.g. a password)
2. Something a user *has* (e.g. an ID card)
3. Something a user *is* (e.g. a fingerprint)

The first two items are straightforward to verify and have achieved common use in password and "smart badge" authentication schemes. The third item has been in limited use for thirty years but has not been widely deployed due to financial and algorithmic problems as well as social resistance.

Each of these three items has strengths and weaknesses for authentication purposes. All three require sensors for gathering relevant information from the person being authenticated and databases of pre-registered information for verifying the <person, information> pair. The first two involve physical or logical secrets, making them precise and easy to verify — either one has the required item or one does not. However, both are also subject to theft and other identity transfer problems, since they are not directly associated with their owners in any unforgeable way. The third, often quantified with biometrics, can not be stolen or transferred easily. However, biometrics differ from the other items in several ways; thus, different approaches are required in order to integrate them into computer systems. The following six differences comprise the main theme of this paper:

**1. Biometrics are not secrets.** Traditional authentication confirms identity based on a physical or logical secret held by the authorized user. The particular algorithm used for verifying these secrets is not assumed to be private and may even be available publicly in source form (e.g., for Linux or FreeBSD). The majority of features acceptable for use in biometric authentication are not secret; instead, they are publicly available by simple observation of the person in question. Thus, anyone can obtain information about a person's features and try to trick the system. Of course, intruders can try to use captured features to trick sensors, but this becomes increasingly difficult as the sensors become increasingly precise. Of more concern is that intruders can compute biometrics directly via open algorithms and deliver them to the system via open interfaces [5], bypassing the sensors entirely. Because biometrics are not secrets, steps must be taken to ensure that they actually correspond to the person being authenticated.

**2. Biometrics are not completely accurate.** When a user types a password, it is encrypted and compared on a byte-by-byte basis with the encrypted password stored in the system's password database for the username being authenticated. The results of the comparison is a boolean: true or false. Similarly, token-based authenticaton schemes check for the presence of a token; without the token, access is denied. Evaluation of biometric characteristics can not produce such clear results, because of normal variations in measured features and measurement environments. For example, variations in facial expression, facial hair growth, lighting, and background all affect face biometrics computed from video images. Therefore, these variations also affect the comparison of measured biometrics to the value associated with the person's digital persona. The result of this comparison is a *closeness of match*, rather than a boolean answer; the closer the match, the greater the confidence that the user is authentic. Perfect matches, and thus 100% confidences, are rare, requiring biometric authentication systems to choose an acceptable level of confidence and cope with the consequences.

**3. Biometrics can be continuously monitored.** The significant user-attention required for password-based authentication has led computer systems to authenticate users only at the beginnings of sessions. Users are expected to explicitly end their session when they leave the computer interface. If they do not, other people can take the place and the identity of the originally-authenticated user. (Some "screen saver" applications require users to re-authenticate after lengthy periods of idle time to partially reduce this threat.) Many biometrics (e.g., face images) can be repeatedly checked without interfering with user activity or attention. This fact can allow systems to replace current one-time authentication schemes with near-continuous verification of user identity, with a corresponding increase in user authenticity [28]. Token-based schemes can also provide this benefit, though any decoupling of the token from the person can result in authentication tokens left at computer interfaces.

**4. Biometrics are expensive to compute.** The signal processing and pattern recognition algorithms involved with computation and comparison of biometrics require much more computation than password or token verification. For one-time authentication, the corresponding computation latencies are often not a major concern; for continuous authentication, this concern is more substantial. In the literature of the biometric community, little detail is given about the time and equipment used to process images [1, 15, 25], and there are very few detailed evaluations [12]; instead, the focus is on improving accuracies. For real system integration, however, the computational costs must be addressed or at least anticipated.

**5. Biometrics are unique per-individual measures.** In theory, it is desirable for the system to ensure that the person using the system is the one authenticated for the given session. In practice, there are circumstances in which sharing of an authentication session or userID are accepted practices. For example, when asking a friend or system administrator for help with a tricky configuration procedure, a user may allow another person to operate within their authenticated session. Continuous biometric authentication would note this change and disable the session. Similarly, shared `Administrator` accounts (e.g., "root") are common for multi-person system administrator staffs. Support for such user conveniences must be considered when integrating biometrics into computer system authentication.

**6. Biometrics are not universally desirable.** Person authentication is an important aspect of computer security, and biometrics can improve the confidence of person authentication. However, there are manifest reasons to protect the privacy of biometrics [14, 34] social worries about "Big Brother" watching our every move Some of the privacy concerns, such as linking independent anonymous personas owned by a given individual, can be partially addressed via improved technical mechanisms. However, the social worries cannot be eliminated, because the potential for such privacy abuses do exist. We do not promote biometric authentication as the right approach for all environments; such is not the case [35]. Rather, our goal is to provide guidelines for more effective integration of biometric authentication, so that increased security and minimized privacy risks are realized when biometrics are determined to be appropriate.

# 3 Designing Biometric Authentication Systems

Most existing systems that augment authentication with biometrics treat them like passwords, creating a variety of security and privacy holes. This section considers more appropriate design choices for dealing with each of the six issues raised in the previous section.

## 3.1 Biometrics are not secrets

Given that biometrics are not secrets, any design must consider how to deal with attempts to fool the sensors by presenting them with reproductions of valid biometric data. For example, a fingerprint lifted from a doorknob can be used as a template for building a three dimensional reproduction that could fool a fingerprint scanner. Likewise, a face image downloaded from the web can be printed and held in front of a camera performing face verification. Systems designers must keep in mind that, since the data they are measuring is public, it will be possible for anyone with sufficient resources to reconstruct whatever biometrics are being measured to fool the evaluation system. Fortunately, sensor technology has become difficult to fool in this way [2].

However, fooling the sensors by presenting them with reconstructions of valid data may not be necessary. An attacker can bypass the sensor entirely by simply feeding valid, recorded data directly to the evaluation system. For instance, a camera that records face images may be attached to its evaluation system through a standard RCA or co-axial cable connector. An attacker could surreptitiously record video of an authorized user and later replay it from a VCR into the interface of the evaluation system to gain access. Therefore, steps need to be taken to ensure that the evaluation system is communicating with a particular sensor and that the readings from that sensor are not manipulated as they are transmitted.

To address this problem, a biometric-based authentication system must ensure that evaluated biometrics originate at a trusted sensor. The system must also ensure the accuracy and timeliness of data used and transmitted to the system where access control decisions are made. The interfaces must be made so that a consumer trusts that the producer of data has not been manipulated to present forged information. In considering the choices before us in forming trusted associations between a camera unit and protected system, guidelines can be found in [31], where security concerns of ad-hoc wireless networks are presented.

One approach is to physically connect all trusted system components in a tamper-proof fashion. For modular implementations, however, this is not a practical approach. Those parts of a system that record, store, and transmit valuable information can be protected through means of both physical security and cryptography. Physical security for sensors is necessary to protect against tampering with their ability to make correct measurements. Any information processing units and storage of sensitive data also needs to be physically secure to prevent processing results from being manipulated. Such security is also necessary to to protect from retrieval of secrets by unauthorized individuals.

Communication with external entities must be authentic, unmodified, and secret in some cases. Authentication is necessary so that the parties involved in a message exchange are mutually assured of the identity of whoever they are communicating with. Integrity is necessary to be assured that messages are not altered in transit. Preventing anyone from knowing what data is being communicated is the realm of confidentiality.

For the case of biometric authentication devices, the location of the various components allow different combinations of physical security, authenticity, integrity and confidentiality. Knowing that sensor data is subject to spoofing, it must either be protected by enclosing the sensor within the same container as the device that evaluates the sensor output or the communication between the sensor and the evaluator must be authenticated, prevented from being modified in transit, and encrypted to protect the privacy of whoever is being sensed. Similarly, evaluation units must be protected from malicious modification so that access control decisions based on their output are assured of the timeliness and accuracy of the evaluations. The messages sent from the evaluator to the access control unit must also be protected to ensure that they are authentic, unmodified, and possibly even secret. Based on how the evaluator is making decisions regarding the measurements it receives, it may not be necessary to fully encrypt the messages between it and the access control entity – the authenticity and integrity of messages may be more important than preventing their content from being observed. Additionally, strong cryptography may be too computationally expensive to require of the evaluators.

### 3.2 Biometrics are not completely accurate

While humans generally do not have any trouble differentiating between people, training a computer to know one person from another has not been a trivial task. Since biometric comparisons result in only a "closeness of match," the access control portion of a biometric authentication system must be set to accept the identity of users at some threshold. This threshold is established as a function of the accuracy of the biometric system and the tolerance of the system administrators to false rejections (locking out a legitimate user) and false acceptances (allowing impostors to have access). The relationship between false rejection and false acceptance as the threshold is changed can be represented on a Receiver Operator Characteristic (ROC) curve, as shown in Figure 3. Each point on the ROC curve corresponds to an underlying threshold value that produces the false acceptance and false rejection rates that can be read from the axes. Over the years, as algorithms have improved, false acceptance ratios and false rejection ratios have fallen (compare, for example, improvement in keystoke identification algorithms between [10] and [22]). Since the result of evaluating a biometric is only a closeness or confidence, with permission granted on measures passing a threshold, an attacker only needs to make sure that they can satisfy the threshold set for the victim system.

The threshold for authentication is not the only variable parameter of concern when constructing a biometric authentication system and optimizing accuracy of biometric match calculations. Sensor fusion between multiple sensors can be performed with matches weighted by the confidence in those various sensors. Such fused biometric evaluation of users has been applied in many situations attaining lower false accept and false reject rates [1, 3].

Alternatively, instead of using multiple sensors, multiple consecutive readings from a single sensor can be applied to increase confidence in the resulting match calculations. The variation inherent in the changing placement of a finger on a scanner surface or the location of a face in a camera frame presents an approximation of multiple independent images for evaluation. Simply accepting an average of multiple readings as the ultimate value to compare with the acceptance threshold can serve to stabilize a reading and give greater confidence in its accuracy.

### 3.3 Biometrics can be continuously monitored

Some biometrics can be continuously monitored while the subjects under surveillance are conducting their normal activities. Any biometric for which this is possible must not intrude on the normal activities of its subjects and must be continuously available for monitoring. Based on the samples that are collected, whatever system is using the biometrics can decide how best to handle failed authentication attempts.

For a biometric to be unobtrusively monitored, whatever means are used for collecting the biometric data cannot require user intervention or cause disruption of normal activities. For example, a sensor that requires contact with a measurable feature is unacceptable for continuous monitoring (e.g. fingerprint must make contact with a reader to sensed). Similarly, if users must change their routine habits so that the biometric sensor can acquire acceptable images, the benefits of periodic biometric checks may be outweighed by the loss of user productivity (e.g. users having to look into a retinal scanner while working and thus hindering their view of the monitor).

Since even the least obtrusive and sample-rich continuous biometric system will occasionally produce failed user authentication checks, the system using the results of those checks has some freedom in deciding how to handle those failures. In this situation the decision-making component knows that it is only a very short time until the next authentication check is performed and can thus base its decision on a history of results rather than just one reading. Therefore, it may be convenient for the system to tolerate a certain number of failed authentications in a series, anticipating that a user has temporarily diverted their attention and successful authentications will resume.

### 3.4 Biometrics are expensive to compute

The design choices for the system center around where processing occurs: on the processor of the protected system or in some dedicated coprocessor. In order to allow the evaluation algorithms to run on the main system, a designer must realize that less processor time will be available for user programs to execute. When they are run on a coprocessor, additional steps may be needed to protect the communication of results and sensor data within the system.

### 3.5 Biometrics are unique per-individual measures

Since each person is different, and within computing there are situations where it is convenient for one person to masquerade as another (e.g. `su root`), there needs to be a way for temporary identity transfers to occur when using a continuous biometric authentication system. This situation can be paralleled in the biometric sense of continuous authentication by simply changing the mapping of whose biometrics are matched against when checking data collected from the sensors. However, allowing one user to sit at a console and perform this re-mapping action while *another* user is expected to be present poses an interesting problem. A solution would be to switch the authentication system into a neutral state and allow the new user to start a session from which they can appropriate other users' sessions once they have proven a claim to a supervisory privilege level.

### 3.6 Biometrics are not universally desirable

Since everyone is different, it is possible to track individuals based on their biometrics. By basing tracking on biometrics, an invasion of privacy is possible. To keep this invasion at bay, biometrics must be protected at every step in a system where they are used for identification. This protection must extend from the sensors capturing the biometrics, through the system evaluating them, to wherever a decision is made based on an evaluation. This requires that multiple aspects of system security be considered. Sensors must be trusted to acquire correct images. Evaluation units must be tamper evident so that physical attacks can be detected. Databases of biometric information must only return information to appropriate authorities. Communications between all the units must be cryptographically protected so that the origin of, as well as information in, an evaluation are known to be correct.

The application of biometrics to person identification is not appropriate in all cases. For instances where exact identification is called for, care must be taken to ensure that biometrics are not divulged incorrectly. For example, databases of biometric data can be protected physically or cryptographically. Biometrics can be used to unlock other secrets or augment them.

Complementing the global encryption of biometric data, any entries in a public biometric database could be encrypted with a password known only to the owner of the biometric. This would allow the owner of the biometric to maintain control over when and where the biometric is used, since it could only be evaluated successfully if the correct password is presented.

Biometrics can be used to hold secrets within a personal "biometric safe." Private keys held by a person, and their associated public key certificates, stored in such a manner allow a level of anonymity to be maintained. The certificates can be distributed as needed to verify the individuality and capabilites of the private key holder in anonymous transactions. With the private keys only accessible through the biometric safe, the owner of those keys can be certain that no one is reproducing their identity. The other party involved in the transaction is assured by the certifying authority issuing the public key certificates of the validity of the claimed capabilities. Thus the identity of the individual using private keys and certificates from their biometric safe never needs to be divulged.

Similarly, a biometric used in addition to a password can enhance the security of an identification step. Passwords combined with keystroke timings has been successfully applied for such a purpose [20]. Using selected stable, invariant features of a person's biometric values and combining them with a secret password leads to considerably more secure authentication. Thus the advantages of "something a user knows" and "something a user is" can be combined to perform authentication that is stronger than either one evaluated separately because of an induced dependency between them.

# 4 Implementation of a Biometric Authentication System

In the interests of testing the guidelines put forward in the previous sections, we have designed and implemented a continuous video based authentication system. The system, illustrated in Figure 1, provides greater confidence in the identity of Linux system users and permits multiple users to have active sessions open on separate virtual consoles. These virtual consoles allow one keyboard, mouse, video card and monitor to serve as input and output devices for multiple, independent console devices that are implemented in software and switched between through the use of keystroke combinations. Images from the camera are constantly being captured by the signal processing system. The largest face detected within those images is selected for tracking. Tracked faces are compared to particular biometric database entries, stored in the signal processing system, as queries from the protected system are received. The biometric database consists of a mapping between usernames and eigenvector representations of user's faces derived from 60 images of the user taken from a camera resting on top of the monitor of the protected system during an enrollment period. The protected system executes an authentication daemon at boot that monitors the system consoles for logins, as well as performing periodic authentications for whatever console is active. The remainder of this section shows how our system addresses the design issues put forth in this paper, describes the system setup and boot procedure, and explains the manner in which users are authenticated on the system virtual consoles, and how console switches, continuous authentication, and biometric hand-offs are performed.

## 4.1 Addressing the design issues

**1. Biometrics are not secrets.** Our system protects communications between the camera and the protected system with symmetric key message authentication codes. The messages that are sent to the protected system contain no biometric data but only communicate a closeness of match. They also contain nonces to prevent replay attacks. A level of tamper-proofness is assumed to be reasonable for protecting the trusted camera's internal communication and storage. Thus, our

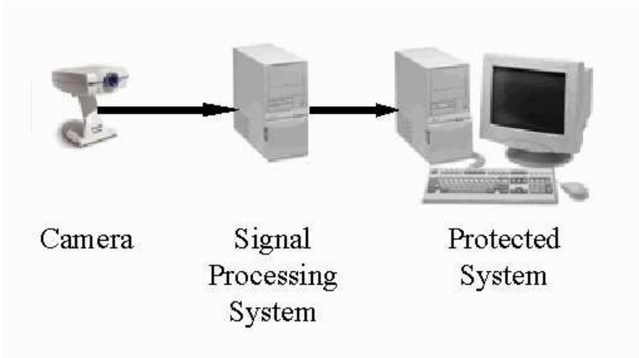


Figure 1: Overview of Continuous Video Enhanced Authentication System

system ensures authenticity and integrity of messages between the trusted camera and the protected system.

**2. Biometrics are not completely accurate.** Knowing that the accuracy of our biometric authentication system is highly sensitive, we keep a history of authentication on which to base our acceptance or rejection decisions. The signal processing system maintains a moving average of closeness of match values, updated as frames are processed. The protected system, when it queries the camera system for identification of a user, adds the received closeness of match to its history of values. If enough of the values found in the history have insufficient confidence to pass the acceptance threshold, then access is blocked until the user can be reauthenticated.

**3. Biometrics can be continuously monitored.** In the interests of not inconveniencing system users and making every effort to continuously collect sensor data, we chose to use face recognition for authentication checks. The view of the camera extends from atop the monitor of the protected system so as to always have any active user in its field of view without subjecting them to physical contact with a sensor.

**4. Biometrics are expensive to compute.** We found that the computational costs of performing the biometric evaluation algorithms placed too much of a load on a processor that would be used for anything other than executing those algorithms. Therefore, a breakdown of our system into a system for signal processing, that analyzes the images, and a protected system, that executes user programs, was implemented.

**5. Biometrics are unique per-individual measures.** To accommodate the needs of users to share sessions and administrators to take over sessions, we implemented a secure console that presents options for changing the biometric identity associated with a console.

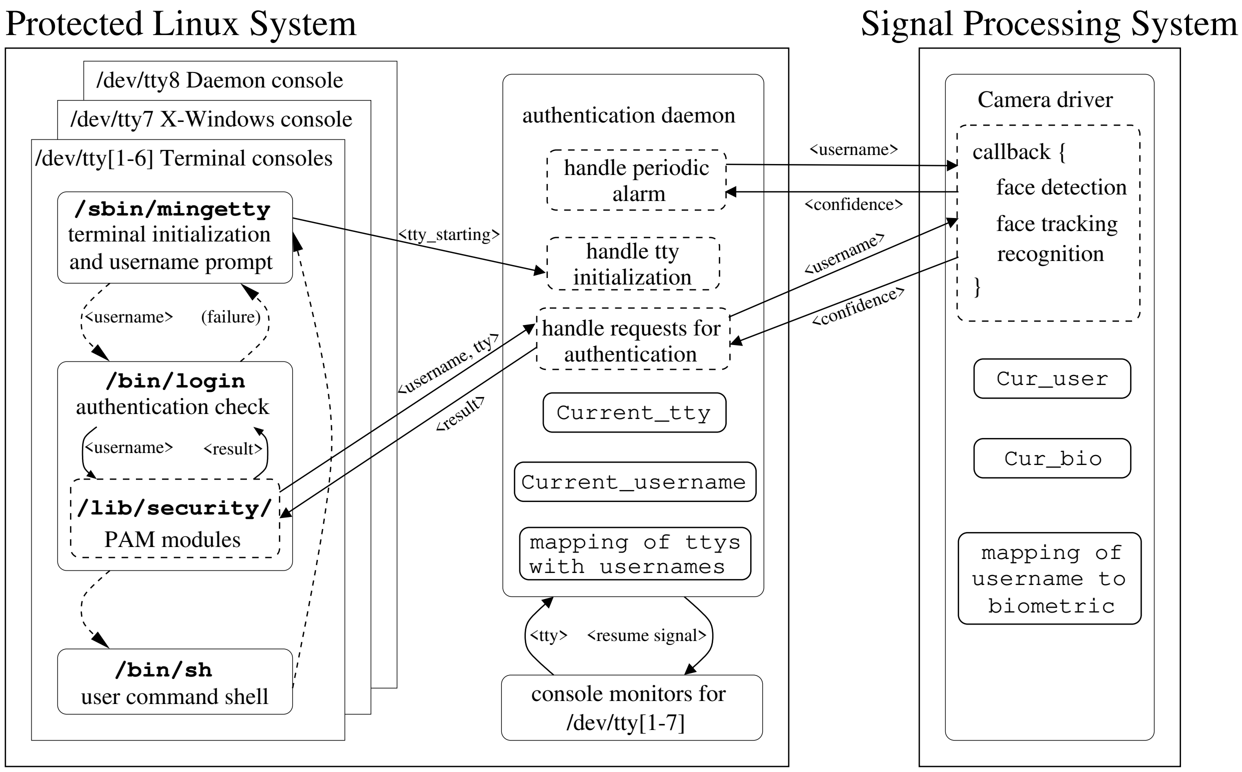


Figure 2: Implementation of Continuous Video Enhanced Authentication System

**6. Biometrics are not universally desirable.** The biometric information gathered in our model is not allowed to leave the camera system. Only measures of confidence regarding user identity are transmitted. Thus, so long as that system remains physically secure, there are no chances for abuse of biometric information to occur.

## 4.2 System boot

As the protected system boots, its initialization routines start the authentication daemon prior to setting up login capabilities on the system virtual consoles. The daemon sets up communication sockets that it listens to for various status updates and requests for authentication. Processes, one for each virtual console, are forked that monitor the consoles for activity. The overall system architecture is shown in Figure 2.

The communication sockets are of three types, one that listens for terminal reset messages sent by a specially modified `/sbin/mingetty` program running on each console, another that waits for authentication requests to arrive from `/bin/login`, and a third for message exchange between the signal processing and camera systems.

Terminal reset messages are sent to the authentication daemon as notification that a user session has terminated on a virtual console. Upon receiving such messages, the authentication daemon knows that the sending console no longer has a user and has been reset to allow for new logins. Therefore, the authentication daemon can stop performing authentication checks should that console become active and can erase the user entry and history that it had maintained for that console.

Authentication request messages are generated for initial console authentication, as described in Section 4.3, and periodic identity checks to enact continuous authentication described in Section 4.4. Initial authentication sends these messages from within a Linux Pluggable Authentication Module (PAM), described in [19, 26, 29], invoked by the `/bin/login` program. This program uses PAM to authenticate users, according to administrator defined policies, using dynamically loaded functions stored in the `/lib/security/` directory. Periodic identity checks are formed within the authentication daemon and forwarded to the signal processing system.

The communication channel between the signal processing system and the protected system is established as soon as the authentication daemon starts. Once connected, the TCP/IP socket is not used until an authentication needs to be performed. Messages sent along this channel include nonces and MD5 Message Authentication Codes (MACs) using a shared secret key to protect their integrity and authenticity. The key material is installed on both machines by an administrator; no provisions for key distribution are made or assumed.

The forked processes for monitoring virtual console activation communicate with their parent, the authentication daemon, through a pipe. These processes each monitor one of the protected system's virtual consoles using the `ioctl()` system call with arguments of `VTWAITACTIVE`, a constant, and a file descriptor, opened on the console to be monitored. This call blocks until the specified console becomes active. When its monitored console is activated, the process informs the authentication daemon of this fact by writing the raw tty device number for the activated virtual console on the pipe. The monitoring process then pauses, waiting for a signal from the authentication daemon informing it to resume its watch for console activation.

### 4.3 Console authentication

The protected system presents users with a standard login prompt on each console where they can be authenticated. The login process proceeds as follows. The `/sbin/mingetty` program collects the username that a user types and passes it to `/bin/login`. The login program activates the PAM function library and takes the appropriate steps to request and verify a password for the account. Then it proceeds to perform a face recognition of the user.

The first step of face recognition requires that the PAM module for face recognition contact the authentication daemon, passing it the <username, tty> tuple. The authentication daemon accepts this information, stores the pairing for use if authentication is successful, and forwards the username, a nonce, and keyed MD5 MAC to the signal processing system. If there is a face currently being tracked, it is compared to the database information stored for the specified user. The algorithm for determining if there is a face in the captured frames is described in [25] and uses neural nets to sequentially scan an image for the presence of faces. The face tracking algorithms are similar to those described in [16, 17] and use the characteristics of human skin color to partition an image into foreground (skin color) and background (all other colors) portions. The face recognition algorithms use the tracked face position and uses the positions of the eyes to resize and straighten the face image. This image is then projected into an eigenspace constructed from the enrollment information. Then the pixel-by-pixel difference between the face image stored in this eigenspace and the current projected image is taken as the closeness of match for the current user. That closeness of match is returned to the protected system, along with the same nonce, to identify the transaction, and a keyed MD5 MAC. When no face is being tracked, a special NULL message is returned. When a user does not exist in the biometric database stored on the signal processing system, an arbitrarily large, and hence invalid, closeness of match is returned in the message.

Upon receiving the response from the signal processing system, the authentication daemon verifies the integrity of the message. The closeness of match metric in a correct message is compared with the administrator defined authentication threshold. A match metric that passes the threshold triggers a successful authentication response to the login module, which allows the user to access the system. The authentication daemon then places the username into the mapping it maintains of which users are active on which consoles and starts a history of match metrics for that user session. Authentications that do not pass the threshold are reported to the login module, which then denies access to the user.

### 4.4 Continuous authentication

Once the user has successfully authenticated to a virtual console, the authentication daemon occasionally checks that the user sitting in front of the protected system is the same as the user who originally authenticated on that console. This action is taken in response to a timer that is triggered periodically whenever the currently active console has a user. The timer generates an `ALARM` signal and the installed signal handler verifies that the current user is still present by contacting the signal processing system. The verification proceeds the same as the initial console authentications, as far as the communications between the signal processing and protected systems are concerned. Once received, the closeness of match is added to the authentication history for the current console. That history is then scanned for an excessive number of failed authentications, and if the failure threshold is exceeded, a console switch is forced. The console that is switched to is controlled by the authentication daemon. It presents the user with a list of the consoles currently in use and allows them to re-authenticate to their own consoles, open a new console, or perform a biometric hand-off as described in Section 4.6.

### 4.5 Console switching

As any user console switch occurs, such as a user changing to another of their authenticated consoles, the processes monitoring the consoles for activity communicate with the authentication daemon. The message consists of the device number for the newly activated console and is sent by the process monitoring that console. The authentication daemon recognizes this console as the new active console and sends a signal to the monitoring process for the previously active console, informing it to resume its wait for console activation. If the users of the previous and current console are different, an immediate identity check with the signal processing system is performed for the user registered as authenticated on the new console and the authentication history is monitored as described in the previous section. Periodic authentications continue at the normal rate.

### 4.6 Biometric hand-off

A user can specify that the biometric data of another user be used in their stead on one of their consoles, and administrators can take over access to the various consoles by specifying that their biometric data be used. This facility is provided through the console that is controlled by the authentication daemon. A menu is presented on that console with options to perform each of these handoffs, as well as authenticate to a new console. When a user-to-user handoff is performed, the current user specifies the username associated with the biometric data that should be used for matches on their console. If that user is then successfully authenticated, by password and face recognition, then the biometric user information mapped for that console is changed to the specified individual in the authentication daemon's internal database. An administrator wishing to take over an existing session announces their intentions through the menu options and, upon successful authentication, the biometric mapping for the specified console is changed to match their identity.

Thus, our goal of permitting multiple users to have active sessions open on separate virtual consoles is met. Additionally, we allow users to designate a handoff of the biometric authentication steps to another user, so that operations can take place in their name. Also, superuser access is not denied to any of the user consoles as long as a correct username, password and biometric check for an administrator are present.

## 5 Evaluation

The performance of our continuous authentication system can be analyzed from the viewpoints of users, system administrators, and system designers. Users are most concerned about everyday usability of the system. Administrators are mostly concerned with the ability of the system to discriminate between actual users and impostors. A system designer is concerned with the burden placed on resources within the overall system.

Mimimal impact on the system users is accomplished through unobtrusive sensing and strategic setting of the authentication threshold. The authentication daemon processes input from its various open file descriptors as data becomes available by looping on a `select` system call. The responses to the data commonly involve user input as terminals are switched and any latency that would be introduced by the authentication daemon is hidden by the time spent updating the video display to the new console data. The single packets sent and received for periodic authentications are inconsequential on the user time-scale, even with checks on the message authentication codes. User experience, as determined by a well-chosen authentication threshold, is ideally the same as without continuous authentication. The actual cost of a failed authentication consists of a console switch to the daemon managed console, a re-authentication on that console, and switch back to the original console. The most common reasons for failed authentications are users leaving the computer and distractions causing user attention to be diverted with poor images captured as a result. A poor image with a face turned to the side or glancing to the left or right can be tolerated to a degree, but the pixel-by-pixel comparison to the stored biometric degrades rapidly as the captured faces deviate from those captured during enrollment. Additionally, the face detection routines cannot consistently detect faces with users wearing eyeglasses, which hampered our ability to use some

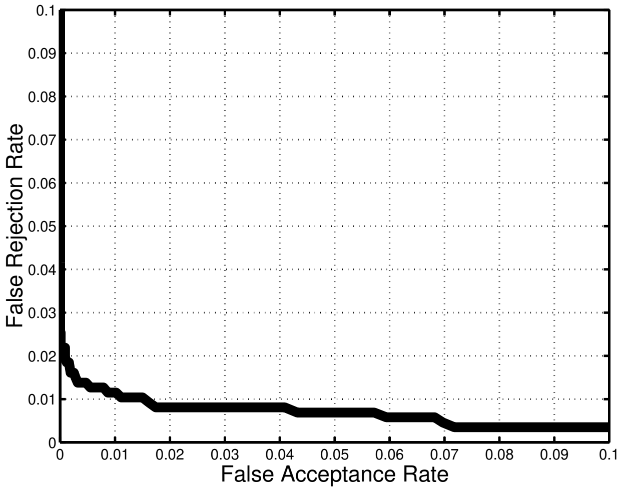


Figure 3: Receiver Operating Characteristic Curve

test subjects. As a consequence of this and due to limited time to perform tests of the system, significant quantifiable results regarding the experience of users of our system are unavailable.

System administrators have the responsibility of securing their systems to the best of their ability, while still allowing users to perform their own work. The greatest impact they can have on a system that uses continuous biometric authentication for access control is through changing the threshold for authentication. This is their means of controlling the false acceptance and false rejection rates. A Receiver Operating Characteristic (ROC) curve displays the trade-off inherent in altering false acceptance at the cost of false rejection, and vice versa. An ROC curve for our face recognition algorithm is shown in Figure 3. The ROC curve was calculated using 51 images acquired from each of 17 test subjects during enrollment. To develop an eigenspace representation of each user face, 50 images are used as training data. The remaining one image was used in testing to generate the false acceptance and false rejection rates. The commonly used leave-one-out testing method allowed us to create as many testing cases as possible. Note that, for our system, authentication thresholds can be chosen that virtually eliminate false acceptance of impostors while maintaining low false rejection rates of valid users.

The optimization of the ROC curve is ultimately the responsibility of the architect of the biometric authentication system. In designing the system, the computational requirements of the algorithms used to evaluate the biometrics must also be considered. The tasks of face detection, tracking and recognition performed during the analysis of video images each evaluated images of different sizes and color depths. Processing was performed on a 550 MHz Pentium-III with 128MB of RAM running Windows NT 4.0 with Service Pack 6. The system contained a Winnov Videum AV PCI video capture card that is connected to a Winnov Color Video Camera. The data acquired by the camera and video capture card consists of 352 by 288 pixel 16-bit color images. The face detection routines operate on a downsampled 176 by 144 pixel 8-bit grayscale images and can analyze 1.12 frames per second with a 94% load on the processor. The face tracking uses full 352 by 288 pixel 16-bit color images and tracks faces at a rate of 6.25 frames per second with a processor loading of 65%. The face recognition routines use 20 by 20 pixel 8-bit grayscale images that are extracted from the full frame using the tracking information. An average of 4.63 frames per second can be tracked with recognition of faces computed while placing a 76% load on the processor. Overall, these heavy computational costs lead to the decision to put the face acquisition and recognition outside of the system being protected.

# 6 Discussion

Biometric authentication is an inevitable new feature of many future systems. In addition to the OS integration issues explored in this paper, there are a number of additional opportunities and challenges ahead. This section briefly discusses a few.

With cryptographically-trusted sensors, biometric authentication can be extended to distributed and mobile computing environments. This can be done by associating the biometric sensor with specific individuals, much like smart cards with PINs, and having person-specific private keys stored inside the sensor device. Alternately, the biometric sensor can be associated with a more centralized authentication service (e.g., Kerberos [21]), trading signed biometric values for capability-granting tickets.

Biometric sensors are also likely to play an important role in secure ubiquitous computing environments. While "smart badges" are convenient mechanisms for tracking and authenticating users in such environments, they are subject to theft and loss. Combined with the inherent lack of physical security involved with most ubiquitous computing notions (e.g., one can access information without being at one's desk), this tenuous lack of control over the authentication mechanism is dangerous. Trusted biometric sensors included in such environments can complement smart badges and address this deficiency.

An interesting question that we have been asked is whether a challenge-response would allow untrusted biometric sensors to be utilized. For example, challenging the person to make a particular facial expression (e.g., a frown) might allow a remote site to distinguish a real person from a pre-recorded image. While an intriguing notion, the danger here is that technologies for recreating facial expressions from pre-recorded images have progressed almost as fast as technologies for detecting facial expressions [13, 23].

Two interesting concepts that emerge from comparisons of authentication techniques are those of authentication confidence and multi-modal authentication. Rather than simply making a binary decision about a user's authenticity, systems could instead remember the confidence of the authentication and adjust the user's privileges accordingly. Highly-sensitive data and operations might be restricted to users of whose identity the system is highly confident; less-sensitive data might be available to users for whom authentication confidence is above the bar but not stellar. Multi-modal authentication combines multiple sensing modes (e.g., passwords, face recognition, and keystroke timings) to increase identity confidence. So, for example, password-only authentication might let a professor check their e-mail but not allow them to access the grade database. Only with successful password **and** biometric authentication can the grade database be accessed.

Biometric authentication does not remove the ability for individuals to have several distinct digital personas. While each such persona would be associated with the same biometrics, this fact can be strongly hidden when the biometrics are combined with passwords or token-IDs. As discussed in Section 3, the biometrics can be encrypted by the password or can be used to unlock passwords. Connecting the different personas via known biometrics would then require compromising all of their different secrets, rather than a single database. While not perfect protection, it may be sufficient for many circumstances. Further, personas with NULL biometrics can be used when appropriate. Biometrics are a tool, not a requirement.

# 7 Related Work

Our efforts in integrating biometric based authentication into computer systems builds on extensive research into sensing and evaluating biometric data [1, 3, 18]. Many companies now provide biometric authentication devices and software. Consortiums exist for the exchange of ideas [7] and the development of standardized APIs [5]. Unfortunately, most current offerings do not consider one or more of the issues expounded in this paper (particularly, the "biometrics are not secrets" issue), making this work timely and important.

Of the existing sensor products available for integration into continuous authentication systems, reachable from [6, 9], very few implement any security features. Those companies that do build products with cryptographic support and tamper resistance, such as [24, 27, 32, 33], provide few details to the public regarding the extent of their efforts. However, whatever measures they are taking are improvements over the products that give no regard to securing the integrity of sensor data or securing the authenticity of communicating partners.

We found only one example of continuous authentication in the literature. Keystroke monitoring for user identification, as documented in [28], tracked the duration of and interval between keystrokes, and reported statistics in terms of mean and variance for users. It may be that continuous evaluation of more complex biometrics is only now becoming possible. The rapid pace of processor development is allowing demanding biometric evaluation algorithms to calculate over larger data sets in less time. Reducing computational complexity does not appear to be a concern of biometrics research groups, rather their focus is on reducing the false acceptance and rejection ratios. The mature algorithms developed over the past three decades, when coupled with the processors of today, may make continuous biometric authentication a reality.

There are other ways for attackers to gain access to systems. If an attacker only needs to place a boot disk in the floppy drive as a system boots up to gain access to the contents of the hard drive, elaborate authentication means are practically worthless. Similarly, having BIOS passwords that can be bypassed by re-flashing the BIOS or by shorting a couple of pins on a motherboard presents an avenue for system access to an attacker. Any attacker with this level of access to a system would not even have to go through such troubles, they could just remove the hard drive and walk away with it to inspect its contents at their leisure. A truly secure system must be protected from system boot throughout each and every user session that is established [4].

# 8 Conclusions

This paper outlines design challenges and options related to effective incorporation of biometric-based authentication into systems. Most current systems violate one or more of our guidelines, resulting in systems that are vulnerable to fairly straightforward intrusion. Specifically, biometrics are not secrets; if systems do not trace the authenticity of biometric data back to a trusted sensor, then intruders can trick the system with easily-obtained biometric information.

A biometric-enhanced authentication system that increases desktop security is presented. A Linux-based implementation of this system is described and evaluated. By using a trusted camera, message authentication codes, and continuous authentication, the effort required to trick the authentication system is significantly increased.

## References


[1] *Second Intl. Conf. on Audio and Video-based Biometric Person Authentication*, 22-23 March 1999.

[2] Biometric Security Body Language. *Laptop Buyer's Guide and Handbook*, pages 92-100, April 2000.

[3] Biometrics. *IEEE Computer*, February 2000.

[4] William A. Arbaugh, David J. Farber, and Jonathan M. Smith. A secure and reliable bootstrap architecture. *IEEE Symposium on Security and Privacy* (Oakland, CA), pages 65–71, 4–7 May 1997.

[5] BioAPI Consortium. http://www.bioapi.org/.

[6] The Biometric Digest. http://webusers.anet-stl.com/˜wrogers/biometrics/.

[7] Biometric Consortium. http://www.biometrics.org/.

[8] Gerrit Bleumer. *Biometric yet privacy protecting person authentication.* TR 98.1.1. AT&T Labs-Research, 1998.

[9] Secure Computing Magazine: Body Parts, February, 2000. http://www.westcoast.com/securecomputing/2000_02/cover/cover.html.

[10] Marcus Brown and Samuel Joe Rogers. User identification via keystroke characteristics of typed names using neural networks. *International Journal on Man-Machine Studies*, **39**(6):999–1014, December 1993.

[11] Michael Burrows, MartínAbadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems*, **8**(1):18–36, February 1990.

[12] Jeffrey Gilbert and Woodward Yang. A real-time face recognition system using custom VLSI hardware. *1993 Computer Architectures for Machine Perception.* (New Orleans, LA), pages 58–66, 15–17 December 1993.

[13] Brian Guenter, Cindy Grimm, Daniel Wood, Henrique Malvar, and Fredrick Pighin. Making faces [facial animation]. *SIGGRAPH 98 Conference* (Orlando, FL), pages 55–66, 19–24 July 1998.

[14] B. Gutierrez, J. Van Os, V. Valles, and R. Guillamat. Congenital dermatoglyphic malformations in severe bipolar disorder. *Psychiatry Res*, **78**(3):133-140.

[15] Lin Hong and Anil Jain. Integrating faces and fingerprints for personal identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(12):1295–1307, December 1998.

[16] Fu Jie Huang and Tsuhan Chen. Tracking of Multiple Faces for Human-Computer Interface and Virtual Environments. Submitted to IEEE Int;. Conf. on Multimedia and Expo.

[17] Martin Hunke and Alex Waibel. Face locating and traking for human-computer interaction. *Asilomar Conference on Signals, Systems, and Computers* (Pacific Grove, California), 31 October–2 November, 1998.

[18] Anil Jain, Ruud Bolle, and Sharath Pankanti. *Biometrics: Personal Indentification in Networked Society.* Kluwer Academic Publishers, 1999.

[19] A Linux-PAM Page. http://www.kernel.org/pub/linux/libs/pam/.

[20] Fabian Monrose, Michael K. Reiter, and Susanne Wetzel. Password hardening based on keystroke dynamics. *ACM Conference on Computer and Communications Security* (Kent Ridge Digital Labs, Singapore, November 2–4). Published as *Proceedings of ACM Conference on Computer and Communications Security*, pages 73–82. ACM Press, November 1999.

[21] B. Clifford Neuman and Theodore Ts'o. Kerberos: an authentication service for computer networks. *IEEE Communications*, **32**(9):33–38, September 1994.

[22] M. S. Obaidat and Balqies Sadoun. Verification of computer users using keystroke dynamics. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **27**(2):261–269, April 1997.

[23] Frederic Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, and David H. Salesin. Synthesizing realistic facial expressions from photographs. *SIGGRAPH 98: 25th International Conferenceon Computer Graphics and Interactive Techniques* (Orlando, FL), pages 75–84, 9–24 July 1998.

[24] Precise Biometrics: Precise 100. http://www.precisebiometrics.com/products/index.html.

[25] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Rotation invariant neural network-based face detection. *IEEE Conference on Computer Vision and Pattern Recognition* (Santa Barbara, California), 23–25 June, 1998.

[26] V. Samar. Unified login with pluggable authentication modules (PAM). *3rd ACM Conference on Computer and Communications Security*, pages 1-10, March 1996.

[27] Sensar: Secure Cam Model C2. http://www.sensar.com/products/products.htm.

[28] S. J. Shepherd. Continuous authentication by analysis of keyboard typing characteristics. *European Convention on Security and Detection*, pages 111-114, May 1995.

[29] Pluggable Authentication Modules (PAM). http://www.sun.com/software/solaris/pam/.

[30] Eugene Spafford. Observing reusable password choices. *UNIX Security Symposium III* (Baltimore, MD,), pages 299–312, 14–16 Sept. 1992.

[31] Frank Stajano and Ross Anderson. The resurrecting duckling: security issues for ad-hoc wireless networks. *International Workshop on Security Protocols* (Cambridge, United Kingdom), 19–21 April 1999.

[32] Mytec Technologies: Touchstone Pro. http://www.mytec.com/products/touchstone/.

[33] digitalPersona: U.are.U. http://www.digitalpersona.com/html/technology/security.htm.

[34] D. Weinstein, D. Diforio, J Schiffman, E. Walker, and R. Bonsal. Minor physical anomalies, dermatoglyphic asymmetries, and cortisol level in adolescents with schizotypal personality disorder. *Am Journal of Psychiatry*, **156**(4):617-623.

[35] John D. Woodward. Biometrics: privacy's foe or privacy's friend? **85**(9):1480–1492. *Proceedings of the IEEE*, Sept. 1997.