Designing a secure biometric identification system for Israel

Ido Efrati, Jesika Haria, Michael Sanders, Xiao Meng Zhang

May 14, 2014

Abstract

Israel has proposed a new biometric identification system, designed to be fully deployed throughout the country. To our knowledge, its design has not been fully analyzed by a group external to the Israeli government. We evaluate the current design, determine weaknesses and vulnerabilities, and propose a new, better system that we believe meets the objectives of the proposal.

1 Introduction

In the present socio-economic environment, people are increasingly more and more dependent on electronic means of payment, communication, verification and interactions. In turn, this pressures political and governmental systems to increasingly provide more forms of electronic verification and authentication. Without identity documents, individuals often cannot exercise basic rights and access services necessary for financial and physical security, formal employment, or democratic participation. Governments and donors cannot effectively ensure that funds reach intended beneficiaries.

One of the biggest risks of storing and accessing data on electronic systems is the risk of identity fraud. Identity theft is often committed to facilitate other crimes such as credit card fraud, document fraud, employment fraud or even terrorism, which in turn can affect not only the nation's economy but its security.

Biometric identification is considerably more accurate and secure than traditional methods of identification and authentication, and it provides an auditable trail of transactions. It offers the possibility of including individuals without documentation and can help streamline and facilitate services in remote, underserved locations.

Common biometric modalities include fingerprints, face recognition, iris, voice, signature, and hand geometry. Every individual possesses an entirely unique biometric profile and biometrics can check these measurable behavioral and physiological characteristics, and then store the information for identity verification at a later date.

As a state under perpetual threat, Israel was one of the first countries to adopt biometric identification procedures. In recent years, the Israeli government established a special inter- ministerial committee, chaired by the Director of the Interior Ministry, and included members of the Ministries of the Interior, Internal Security, and Justice, the Prime Ministers Office, the Israeli Police, the Israeli Defense Force, and the Israel Airports Authority. Although it served in a largely regulatory role, the committees goal was to hasten the development, legislation, and standardization of biometric identification in Israel.

It proposed that the biometric data on identification cards and passports should match the standards of the United States and European Union, and should follow the technical guidelines of the International Civil Aviation Organization (ICAO), the regulations of the International Organization for Standardization (ISO), and the National Institute of Standards and Technology (NIST). For fingerprinting, the FBI's WSQ Grayscale Specification is widely accepted [2].

After a long period of parliamentary and public debate, the Knesset, the Israeli parliament, passed the "Biometric Law in December 2009. Over a year later (May 2011), the government approved the legislation giving the Interior Ministry authorization to issue "smart ID cards to Israeli citizens. Every citizen who receives a new card is required to provide two fingerprint samples and a digital face photo, which will be stored in the governments biometric database. [7]

In its initial stage, the smart ID card will allow every citizen to perform tasks such as filling in electronic forms vis-à-vis the government databases, and signing the forms with a digital signature.

2 Current System

As previously mentioned, in 2009, the Israeli Knesset enacted the "the Biometric Database Law". The law defines the necessary arrangements to issue a biometric identification documents, i.e., passports and identification cards, that will enable the identification and authentication of Israeli citizens in a manner that will reduce forgery and identity theft [1]. These biometric identification documents will include a facial features image as well as a derived templates of fingerprint of both forefingers. These features are unique identifiers to each person, and cannot be forged easily. This reduces the possibility of forgery and identity theft.

In January 2013, the Israeli Ministry of the Interior started a two-years pilot, in which Israeli citizens can volunteer to give their biometric information and receive a biometric identification documents. This section will describe the current design of the pilot system, and will analyze it via different scenarios.

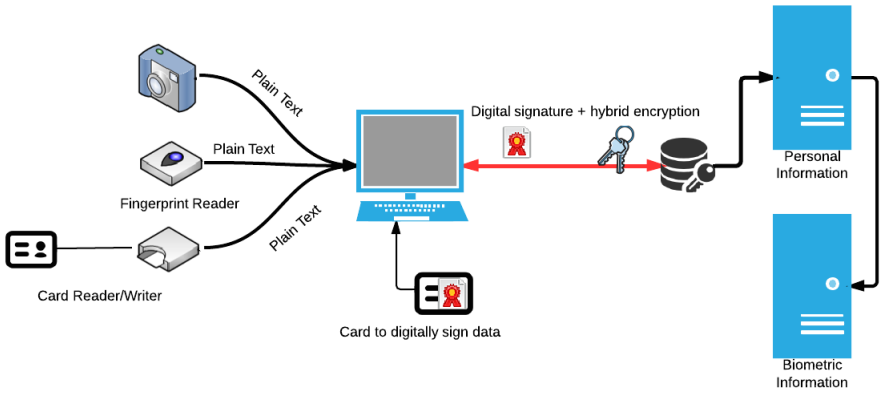

Figure 1: Current design of the Israeli Biometric identification system.

Figure 1 describes the current design of the pilot system. The remaining part of this section will refer to this system.

Issuance of a biometric documentation:

Every Israeli citizen over the age of 16 is required to have an identification card. A citizen may decide to opt into the biometric pilot, if he or she is over 18, or if he or she has a legal guardian's permission, in the case where the citizen is 16 to 18 years old. Alice, an 18 years old Israel citizen, decided to opt into the biometric pilot. Thus, she goes to the Israeli Ministry of the Interior to receive her new biometric identification documents. After Alice verifies her identify via the required documents by law, e.g., birth certificate, parents' IDs and so forth, the clerk will take her fingerprints via the fingerprints reader, and will take her facial picture. Furthermore, Alice will supply additional personal information, e.g., her current address. Afterward, the clerk will have to insert an employee card to his or her station, that will digitally sign the request to create a new biometric identification document, and will submit this request to a central server. It is important to note that the card is preloaded with a key that will allow the clerk to sign the request. Upon submission the biometric data and personal data will encrypted with hybrid encryption and be sent to the servers [6].

An hybrid encryption is an encryption that combines both public-key encryption as well as symmetric-key encryption. Thus the encrypted message gains both the convenience of the public-key encryption, as well as the efficiency of a symmetric-key encryption. In this scheme both the clerk's station and the server generate a symmetric key and encrypt the message with that key. They then use each other's public key to encrypt the symmetric key, and send it over the channel, then they send the encrypted message that could be decrypted with the key that was just shared [9].

Once the request gets to the server the personal information is stored as plain text in the personal information server, and the biometric information is forwarded to the biometric information server and stored there in its encrypted form, along side its corresponding decryption keys. Finally, all of the biometric information that passed through the personal information server is from that server, and resides only in the server side.

Upon a successful identity creation, the encrypted derived biometric templates as well as a unique identifier that relates the information to the decryption key that resides in the server, is burned on Alice's identification documentations. This information will serve Alice in the future in case she would need to authenticate herself.

Authentication with a Biometric Identification Documentation: When Alice would like to get any service that requires an authentication, e.g., crossing the border, she will have to authenticate with her biometric identification documents. Alice will present her documentation and her card would be inserted into a card reader. A valid station with an employee card, i.e., the card with the keys for the digital signatures, will be able to access the biometric templates that were stored on her card. Alice will then have to give "live" biometric sample, i.e., re-scan her fingers, and all of these information would be encrypted, as mentioned above, and would sent to the server. The server will perform an authentication, and will return one of three possible answers:

1. Alice was authenticated successfully
2. Alice was not authenticated, i.e., Alice is not who she claims she is.
3. The quality of the sample was not good enough and there were several samples that matched.

In the last case, the entries would be presented to the agent who authenticate Alice with percentage of matching. The agent would be able to examine the entries and verify the if Alice who she claims she is by checking her picture as well as asking her identification questions that only she would be able to answer; thus, guaranteeing authentication [6].

3 Major design vulnerabilities

The following section will cover the major design security vulnerabilities in the current pilot of the biometric system.

1. **Direct linkage of database entries**
   The personal data is stored on one set of servers, and the biometric data is stored on another set of servers. However, there is a direct one-to-one mapping between the two pieces of data, i.e., Bob's personal information is pointing directly to Bob's biometric information. This means that if any one set of servers/ server is compromised, and a particular entry in the table is known, the attacker can easily find the corresponding piece of data.
   For example, as can be seen in Figure 2, if an attacker manages to

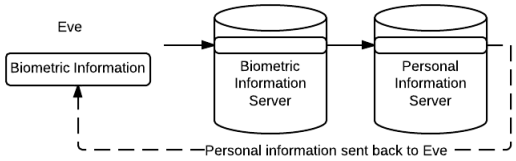

Figure 2: Biometric to personal data linkage

obtain the victim's fingerprints and is able to break in to the system, then he can find out the personal information of the victim since the template derived from the fingerprints that is stored in the biometric database has a direct pointer to the location of the victim's information in the personal information database. This violates the confidentiality of the victim's data seriously.

Conversely, as can be seen in Figure 3, if the attacker has personal

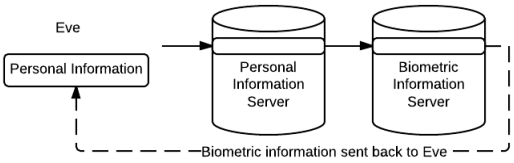

Figure 3: Personal to biometric data linkage

information about a victim and has access to the personal information server, then they can get access to the templates stored in the biometric information, and can potentially use those to their advantage while trying to spoof an authentication with the information servers. Thus, this design currently violates both the confidentiality and authenticity of citizens.

2. **Unsecured and unauthenticated database communications**
   In the current system, communications between the database servers are neither encrypted nor authenticated. Whenever a server communicates with another server, the only type of authentication check performed between the two servers is that the data being requested or sent is that of a valid user. This setup is highly insecure and presents a significant vulnerability in the presence of an active attacker, because

it is very feasible for the attacker to recover the private information of people stored on a given database server.

For example, if an active attacker knows which server holds users' personal information, he can try to construct a fake request (by flipping random bits of a valid request he may have eavesdropped on). If the "fake" request he formulates matches up with a user stored on that database server, then the server will treat the request as one from a trusted server, and return the requested information (which in this case, may be the personal information of a person who happened to be unfortunate enough to match up to the fake request). What makes the situation worse is the fact that the database server will send back to the attacker the requested information without encrypting the data! This means that once an attacker guesses a valid request for a user stored on the database, he will have complete access to all the information stored on the user.

3. **Key strength**
   While not necessarily a vulnerability of the current system, something that needs to be accounted for is that this system is by its nature expected to last several decades. Thus, the encrypted information must be encrypted in such a way to maintain its security for an extended timeframe. Ideally, it would be optimal to choose an encryption scheme that would not require any key changes through the lifetime of the data. However, as computing technology progresses and becomes more powerful, it is likely that current schemes will become obsolete prior to the data becoming irrelevant.

4. **HTTP communication between local stations and central servers**
   The communication between the local stations of the administrative officials and remote servers that store both the personal and biometric information of citizens is performed over HTTP. The rationale used by the Israeli Government is that since the data is encrypted, passing it over an unsecured communication channel would not compromise its security. It must be noted that once the data reaches the remote servers, communication between the servers happens on a private network with special purpose cables dedicated to the communication. This means that the internal communication amongst the servers is reasonably secure. However, since a system is only as strong as its weakest link, the

use of HTTPS in sending requests for personal/ biometric information as well as sending that information back is extremely risky. If a key used for the encryption of any of the information is leaked, then all previous communication encrypted with that key can be leaked. In other words, this system does not allow for perfect forward secrecy.

5. **Physical card security**
   As part of generating a new identity and submitting it to the system, an authorized user (i.e., the clerk operating the station, not the person whose identity is being created) must digitally sign the request. The key for this signature is located on a card with an integrated circuit chip. Prior to submitting the request the clerk inserts their card into a reader, and a signature is generated and submitted with the request. If the key on the card is compromised without the knowledge of the clerk, an adversary could impersonate the clerk on the system if they were able to gain access to a workstation. Thus, this is almost a single point of failure in the current system.

6. **Unsigned data in requests**
   Currently, whenever a person requires an identification card, he will need to go to the Ministry of Interior, provide his biometric data to an Ministry of Interior agent, and have the agent submit a request to the database servers for a new identification using her biometric data collected. However, the problem here is that only the request is verified, rather than the biometric data itself. The verification check on the request only ensures that the request is filed by a valid authority (i.e. actual MoI agent). Figure 4 shows the current setup.

   This setup makes it possible for an adversary to possibly create multiple identities of himself – either through convincing an Ministry of Interior agent that he does not yet exist in the database, or through bribery, or even through stealing a MoI agent's key card used to submit a request himself. Regardless of the case, because of the fact that the database servers approving the requests do not validate the biometric data (such as ensuring that they do not yet exist in the database), it is possible for the same person to submit his biometric data multiple times and hence create multiple identities of himself. Figure 5 shows one case of how the lack of signing data can be exploited by an adversary.

7. **Keys stored alongside encrypted data**

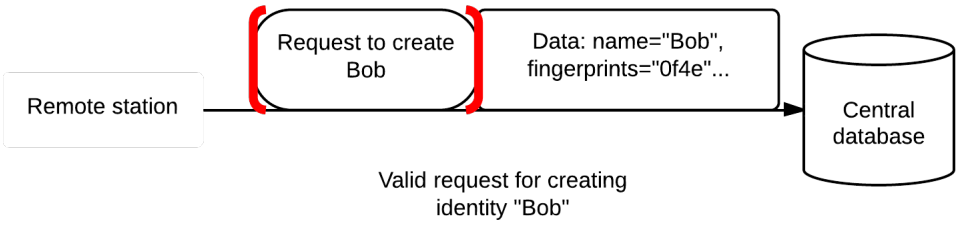

Figure 4: Diagram describing the request. The red brackets indicate the portion of the request that is signed.

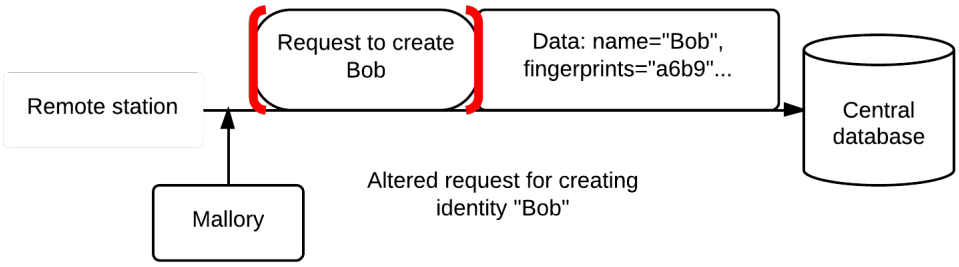

Figure 5: Mallory, an active adversary sitting on the channel, has modified the biometric data contained in the request, but the request is still accepted by the server.

can't say it is terrible idea In this system, the keys for decryption of data are stored on the same server as the data itself. While not necessarily a bad idea, if incorrectly implemented, hacking one server means that not only can the keys be accessed, but also the data itself. Hence, this is effectively equal to storing the data unencrypted.

8. **Physical security of stations**
   Another pressing vulnerability is the lack of physical and hardware security for the database servers that hold sensitive users' information. In 2007, a temporary Ministry of Interior employee decided to copy a portion of the databases of private personal information onto a flashdrive, and later published the flashdrive containing information on many Is-

raeli citizens (including the current Prime Minister of Israel) to the public. The reason that agent was able to accomplish what he did was because there were no physical nor hardware safeguards set up to prevent the agent from stealing all of the sensitive data from the database server. This same problem is present in the biometric ID system, in that there are no physical restrictions in place preventing people from copying data from the various databases onto a flash drive. Given how sensitive biometric information is, it is imperative that restrictions are put in place to prevent attackers from doing what happened back in 2007.

9. **Inability to recover from data leakage**
   As currently implemented, biometric data is stored as itself, with no additional information. The issue with this occurs when some or all data is leaked. Suppose an adversary is able to access someone's biometric data and create an ID card with the adversary's description printed on it, but with the stolen biometric data coded onto the card. When the adversary presents the card to a verifying agent such as at the border, the agent will see the card with the picture and name of the adversary, but when the card is checked the system will see a request to verify the leaked information. Since there is no additional information stored in the database, credentials cannot be revoked. Since people cannot reasonably alter their physical characteristics, this should be changed.

10. **No firewall logging**
    Currently, there are no logs in the firewall of database servers tracking the type and number of requests coming in to the server. This makes it possible for malicious users (who could be anyone, including Ministry of Interior agents themselves) to submit illegal requests without fear of repercussions. Due of the fact that logs are not used anywhere in this system to track requests, it is not possible to trace bad requests to find their originators.

4 Proposed Design

In this section we describe our solutions to each of the vulnerabilities and issues outlined in the previous section. Figure 6 provides a diagram showing

where each point is most relevant, with the numbers corresponding to the points in this section.

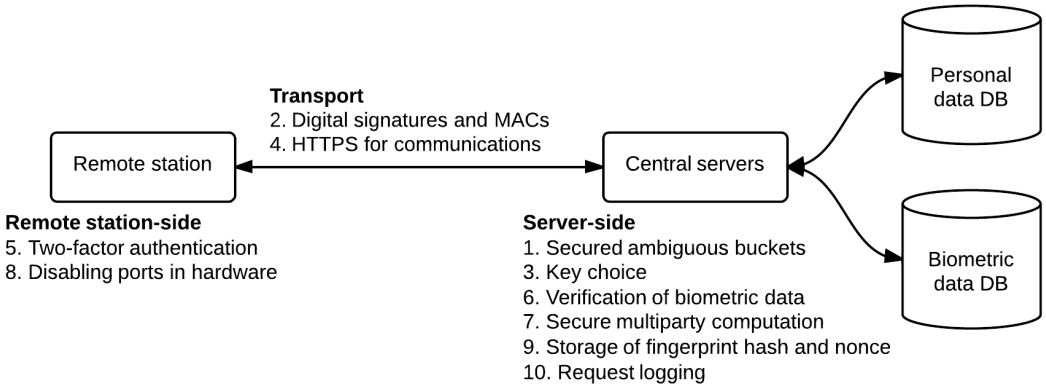

Figure 6: A diagram showing a simplified layout of the new system, with numbers indicating where each design point is applied.

1. **Secured Ambiguous Buckets**
   As mentioned above, one of the main vulnerabilities of the system is the direct link between personal information and biometric information. Thus, if an adversary gains access to the system, he or she can derive either the personal information from the biometric information or vice verse. Since biometric information clearly cannot be replaced once it falls into the hand of the adversary, a solution that prevent such vulnerability is imperative. Thus, we will examine a solution that was partially proposed by Professor Adi Shamir, and will expand it to make it even more secure.

   Professor Shamir suggests splitting each entry in the biometric system into two buckets, the first, for personal information, and the second, for biometric information. It important to note that each of these buckets would contain k entries, but the entries would not be linked to each other. Thus, if an adversary gains access to the server he or she would not be able to discern a relationship between the personal and the biometric information. Furthermore, even if the adversary knows the mapping between the buckets, there are still $k!$ possible

mappings in each pair of buckets. For an example of scale, at $k = 100$, this still amounts to roughly 10^{157} possible mappings. With a CPU at 10^{13} FLOPS, and 10^7 seconds a year, this would still take 137 years to compute, safely larger than the expected lifespan of a human. Furthermore, this also provides citizens with an additional layer of security in that the government officials themselves cannot peek at a citizen's personal information, given that they obtained the biometric information of that individual, or vice versa, unless the citizen himself/ herself was present to provide the complementary information.

Based on Professor Shamir's suggestion, we propose a secured ambiguous buckets design, as follows:

(a) Each personal information entry would be hashed with a pseudo-random deterministic hash function, e.g., SHA-3, to a bucket.

(b) Similarly, every biometric information entry would also be hashed to a bucket.

(c) Collisions would result in doubling of the hash table.

(d) Biometric information would be stored in its hashed form. Thus, it would not be possible to rederive it due to one wayness.

(e) A bucket size of $k = 500$, will require $\frac{Num-Of-Citizens}{k} = \frac{6000000}{500} = 12000$ buckets. This is a feasible size for an efficient data structure that does not add to much overhead.

The above design guarantees security for several reasons. First, there is no direct link between a particular piece of personal information and corresponding piece of biometric information. Second, even if an adversary is gaining access to the biometric information and can find a relation between a personal information and a biometric information, he or she would not be able to use the biometric information because it was hashed. Third, if an adversary knows the mapping between the buckets, he or she still has $500! = 1.2 * 10^{1134}$ possible mappings, which are computationally infeasible to allocate to a particular set of biometric- personal information pair.

Proof that Authentication is still unique: The above change to the system, does not change the process of issuing biometric identification documents. However, one would have to examine if such a

process might cause false authentication/identification due to the removal of the direct linkage between personal information and biometric information. As one may imagine, if both Alice and Bob try to authenticate, with high probability they would be mapped to a different biometric bucket. Furthermore, assuming that they both map to the same bucket, the probability that the hashed data of their fingerprints collide is $\frac{1}{2^{HASH-SIZE}}$. For example, even if we only take the first 40 bits of our hash value, the probability that the hashes of their fingerprints collide is $\frac{1}{2^40} = 9 * 10^{-13}$. Thus, one can see that the unique authentication still holds.

2. **Utilizing digital signatures and MAC protocols**
   In order to keep our new design secure, we propose a two-layer scheme to defend the communication channel against both active and passive attackers. First, we will need all communication between the servers be signed with an RSA digital signature. Whenever a server receives any form of communication from another, he will need to validate the digital signature signed on the message to learn if the message comes from a valid server. This helps prevent the active attacker from being able query a database server and continuously send "fake requests" until one of those requests match up with a valid user stored on the database server (the way he did earlier). Instead, the attacker will also need to construct a valid signature for each request he generates, which will it much harder for him to send fake requests to the serve (that will be processed by the server).

   The second layer that we will impose on our system is that all messages communicated between servers will be encrypted with RSA and will also contain an HMAC message authentication code. This ensures that even if the attacker is a passive listener or succeeds at sending a fake request with a valid digital signature, the response he hears from the server will be RSA encrypted results that he will need to decrypt. Additionally, the HMAC is used to prevent the attacker from being able to modify the results that a database server send outs and convince other servers that the results are valid.

   Here, we use the concept of "defense in layers" to make it even more difficult for an attacker to recover private information from a database server.

3. **Key choice**
   To solve the issue protecting data in the long term, the best option we see is to use long keys currently available with a secure cipher system. Currently, that would be AES with a 128-bit key, which is approved for use with US documents classified at the secret level. To give a sense of the timescale needed to crack a 128-bit key, we assume we have a computer capable of executing 10^{15} operations per second, and each AES encryption requires 10 operations. Therefore, cracking a 128-bit key would take at least $\frac{2^{128}}{10^{14}} \approx 2^{80}$ seconds, which is ˜36 quadrillion years [5]. That being said, a good idea is to increase the number of rounds used in the AES encryption scheme, which would increase the security of the scheme without requiring substantial overhaul. This idea has been proposed by Bruce Schneier following revelations that attacks on AES were gaining ground (though not to the point of rendering AES insecure) [10].

4. **HTTPS for network communications**
   To ensure perfect forward secrecy, so that compromised keys cannot compromise information previously transmitted across the channel, we should use ephemeral session keys to secure the channel via HTTPS [4]. This not only provides an extra layer of encryption, but also provides for keys in one session being unusable for the next, protecting the confidentiality of data. This can be achieved via an elliptic curve Diffie-Hellman key exchange, which keeps using the private key for authentication, but uses an independent mechanism to agree on a shared secret [8]. The overhead imposed by SSL would be negligible, especially at the rate of the requests coming in. [3]

5. **Complementing ID cards with two-factor authentication**
   To solve the problem of potential card compromise, which would allow an adversary to impersonate an authorized data enterer, two-factor authentication should be used. For this system, passwords are the best choice for the second factor. This satisfies the two needed items: something one knows, and something one has. Normal password security steps should be taken, including enforcing a certain amount of entropy through minimum lengths and required characters, as well as salting passwords to prevent rainbow table attacks [11]. While passwords are certainly not invulnerable, adding this second factor will seriously com-

plicate an adversary's objective of inserting malicious data.

6. **Authentication and verification of biometric data**
   In order to eliminate the problem of allowing adversaries to potentially create multiple identities of themselves, we need to verify and authenticate the biometric data provided by a user to ensure that it does not yet exist in the biometric database. This can be achieved by encrypting the biometric data provided and checking the encryption with the database servers to see that no duplicate copies of the biometric data exist. One potential problem with this additional check is that it may increase the processing time required to create a new ID for a user. However, this is a necessary tradeoff that needs to be made in order to prevent malicious users from being able to create multiple fake identities.

7. **Secure multiparty computation**
   To ensure separation of keys and the data, one could either store them on separate servers, but then we would need links between the servers that would have to be obfuscated in order to provide the same level of security. We could also continue to store the keys on the same server as the data, but in order to secure it, we would have to either separate them in hardware, or perform a virtual client-server software/ logical separation, so that the data is safe even if part of the server is hacked.

8. **Disabling ports in hardware**
   To prevent a repeat of the accident that occurred back in 2007, where a temporary Ministry of Interior employee copied a portion of the databases containing sensitive peroneal information onto a flashdrive and leaked the data to the public, we need to minimize the ability of adversaries to attach physical media to the system. We see three options:

   (a) disable hardware ports in BIOS

   (b) use terminals without ports

   (c) use remote workstations that are just a virtual machine on a central server

   Option (a), while straightforward and simplest to implement, can still be undone by a determined adversary if he is physically in the room

with the workstation. Similarly, option (b) can be defeated by an adversary removing the computer's own hardware, such as the hard drive. Thus, option (c) is the best option for our system. It allows physical security aspects to be focused on the central server facilities and minimizes the attack surface on the remote workstations. It should be noted that this solution is not as secure against a systems administrator or other agent operating the central server facilities. However, this can be mitigated by requiring someone wishing to access the server facilities to be accompanied by someone else.

9. **Store hash of fingerprints and nonce**
   We can solve two issues by storing hashes of fingerprints along with a nonce in the database, instead of the fingerprint vector itself. By storing a cryptographic hash of the fingerprint vector, information about the vector itself can't be determined if the database is leaked in unencrypted form. By storing a nonce, it now becomes possible for a set of credentials to be revoked. If a set of biometric data is compromised, and the compromise is known, people whose data was leaked can get new identity cards made, with a new nonce. An adversary who attempts to present the compromised data with the old nonce will have their attempt rejected, as the system will compare both the fingerprint vector hash and the nonce.

10. **Log all requests**
    An additional check that can be added to our system is to maintain a log of all the requests submitted by the Ministry of Internal employees. This log only records the type of request submitted (not the actual data or results), a timestamp of when the request was initiated, and who initiated the request. This log does not add any extra layers of security to our system in the sense that it does not physically prevent malicious users (including Ministry officials) from attacking (or stealing data from) the system. However, having this log in place will make it easier to identify the perpetrator of malicious requests or attacks on the system, including authorized users who are committing malicious acts. We will use the log as a deterrence mechanism rather than a defense mechanism, with the expectation that an adversary's fear of getting caught will deter them from committing the crime. As a rough estimate on size, we estimate that one log entry will require 500 bytes,

and approximately 500,000 log entries will be generated per day. Using those numbers, we estimate that approximately 90 GB of data will be generated yearly, which is very manageable for a system of this scale. The length of time for which logs will be retained should be manageable up to a couple years, at the very least.

5 Preventing Identity Theft

One, if not the most important, invariant that our system has to preserve, is the prevention of dual identity creation, i.e., a user with two identities, as well as identity theft. In order to verify that our system preserves this invariant, we will perform a case analysis.

1. **Eve would like to steal Alice's identity**

   Eve forges all of the required documentation, e.g., birth certificate, and shows up at the Ministry of Interior before Alice has a chance to associate her biometric information to her identity. If Eve's fake documentation passes the clerical inspection, Eve will be assume Alice's identity. However, she will have to give up on her own identity because her fingerprints are already in the system. Furthermore, when Alice arrives to receive her identification, the clerk will notice that her personal information is associated with another identity. This identity has Eve's photo and biometric information, and the clerk will be able to inform the authorities.

2. **Eve was able to steal Alice's identification documentation**
   When Eve tries to authenticate she will have to supply a live fingerprint sample. Thus, her fingerprint template would not match the one of the server and the one on the stolen card. Even in the unlikely event that both Eve's and Alice's fingerprints hash to the same bucket, the unique identification number on Alice's chip will correspond to Alice's photo in the system, which will allow the verifier to visually see that Eve is not Alice.

3. **Eve is trying to receive two different biometric identification documents**
   The first document is Eve's valid document, i.e., her first entry in the

system. Thus, the creation of such document is legal and valid. If Eve then comes back and pretend to be Alice, she will have to give her fingerprints sample again. Since our hashing scheme is deterministic her fingerprints will map to the same bucket, and will have an identical hashing. Thus, the system will notify the clerk that such a biometric ID already exists. On further lawful examination, the clerk will be able to see that these two entries are identical, and Eve's case would be forwarded to the authorities.

6 Limitations

As with all systems, human error cannot be taken out of the equation. Compromising an authorized user who is one of the identity creator clerks is not something that can be easily guarded against, and our system only allows for observation of an event like this, not automated detection or defense. Additionally, our system supposes that there is a feasible way to change encryption schemes for the data in a practical manner in the event current schemes are broken. If this is not the case, it may take an impractical amount of time to switch encryption schemes. Also, our system relies on public-key cryptography, and thus requires public key infrastructure. If the PKI scheme is compromised, the entire system will be compromised.

7 Future Work

Looking ahead to the future, we hope to see an implementation of a biometric identification system that takes into consideration our proposed design. This will allow us to test out our design, explore and search for new vulnerabilities or areas of concern, and examine many of the tradeoffs that we made while formulating this design. Specifically, one of the main decisions made in our design is to use secured ambiguous buckets to tackle the problem of eliminating direct linkage of database entries. Other ideas that we have formulated for this problem include the use of secure multiparty encryption and secret sharing schemes. We believe that it would be a worthwhile investigation to experiment with (and possibly implement) these different ideas, and run a comparison to decide which one is more secure and effective.

8 Conclusion

In this paper, we identify critical vulnerabilities of a proposed Israeli biometric identification system. Based on the stated objectives of the system and our analysis of the vulnerabilities, we propose a design for a new system that should satisfy the current expectations of the biometric ID system. We use several concepts, such as secured ambiguous buckets and hashing biometric data, that have not been combined before to offer a reliably secure system. Notably, this is the first English survey of the currently proposed system that we are aware of.

References

[1] Questions and answers.

[2] WSQ GRAY-SCALE FINGERPRINT IMAGE COMPRESSION SPECIFICATION, Dec. 1997.

[3] How much overhead does SSL impose?, Feb. 2009.

[4] Google online security blog: Protecting data for the long term with forward secrecy, Nov. 2011.

[5] How long does it take to crack DES and AES?, Sept. 2011.

[6] Biometric system pilot protocol, June 2013.

[7] Ben-Yosef, M., and Cohen, E. We'll tell you who you are, Oct. 2011.

[8] Bernat, V. SSL/TLS & perfect forward secrecy, 2011.

[9] Cramer, R., and Shoup, V. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack, Aug. 2003.

[10] Schneier, B. New attack on AES, Aug. 2011.

[11] Steven, J., and Manico, J. Password storage cheat sheet, Apr. 2014.