Formal Security Analysis of Australian E-passport Implementation

Vijayakrishnan P **Josef Pieprzyk** **Huaxiong Wang**

Department of Computing
Macquarie University,
New South Wales 2109,
Email: {krishnan,josef,hwang}@ics.mq.edu.au

Abstract

This paper provides a detailed description of the current Australian e-passport implementation and makes a formal verification using model checking tools CASPER/CSP/FDR. We highlight security issues present in the current e-passport implementation and identify new threats when an e-passport system is integrated with an automated processing systems like *SmartGate*.

Because the current e-passport specification does not provide adequate security goals, to perform a rational security analysis we identify and describe a set of security goals for evaluation of e-passport protocols. Our analysis confirms existing security issues that were previously informally identified and presents weaknesses that exists in the current e-passport implementation.

Keywords: electronic passport, formal methods, model checking.

1 Introduction

For improved security at border control checkpoints, Australia introduced biometric enabled passports in 2005 that conform to the e-passport guideline developed by International Civil Aviation Organisation [ICAO] (ICAO 2005). The e-passport guideline describes the integration of a biometric enabled contactless microchip with Machine Readable Travel Documents [MRTD]. It describes the communication protocol and provides details on establishing a secure communication channel between an e-passport and an e-passport reader. The guideline also describes details regarding storage mechanisms and techniques that should be employed for protection of e-passport data.

The ICAO e-passport guideline uses existing approved standard such as ISO14443, ISO11770, ISO/IEC 7816, ISO 9796, RSA, DSA and ECDSA. But, we were able to identify security weaknesses that still exists in e-passport protocols recommended by ICAO.

1.1 Related Work

The authors in (A Juels et al. 2005) presented some security and privacy issues that apply to e-passports. The contactless chip embedded in an e-passport allows the e-passport contents to be read without direct contact with an e-passport reader and, importantly, with the e-passport booklet closed. The authors raised concerns as to whether data on the chip could therefore be covertly collected by means of "skimming" or "eavesdropping", as the encryption key used in basic access control protocol can be compromised due to low entropy of the key [the length of key being only 56 bits]. In (Gaurav S. Kc & Paul A. Karger 2005), the authors suggested that an e-passport may be vulnerable to "splicing attack", "fake finger attack" and attacks that can be carried out when an e-passport bearer presents the e-passport to hotel clerks. There has also been considerable press coverage (Bobbie Johnson 2006, Will Knight 2006, David Reid 2006) on security weakness in e-passports. These reports indicated that may be possible to "clone" an e-passport.

The "cloning" attack does not compromise border security, as to do that, an attacker should be able to modify the details and still maintain the integrity of the data and its corresponding hashes. However, cloning of a e-passport is a major privacy issue as an attacker would not only be able to obtain the passport bearer's details but also his/her biometric details stored in an e-passport. The risk of eavesdropping is increased by the surveillance environment in which border checks occur, particularly as border control processes become more and more automated, as in Australian *SmartGate* system (Service 2006). This will ultimately assist in covert collection of e-passport data.

This paper presents a formal analysis of the current first generation e-passport protocols. We were able to formally verify that e-passport protocols do not meet basic security goals like data confidentiality, data integrity, key integrity, mutual and data origin authentication and is vulnerable to attacks that would compromise both privacy and security of an e-passport bearer.

1.2 Organisation

In Section 2, we provide a detailed description of Australian e-passport implementation, the focus being on cryptographic protocols. In Section 3 we define our security goals for formal verification of e-passport protocols and present security analysis of the entire protocol suite for the highest level of security as defined by the ICAO guideline. In Section 4 we present our formal verification of the e-passport implementation using CASPER/CSP/FDR. Finally, we conclude in Section 5 with a summary of weakness and recommendations for better e-passport implementation.

2 ICAO E-passport Specification

The ICAO standard, DOC 9303 (ICAO 2005) for MRTD describes a contactless microchip that con-

form to the ISO 14443 (ISO/IEC 2000) embedded within an e-passport booklet. The microchip duplicates data that is recorded in the machine readable zone [MRZ] of an e-passport and information that appears on the e-passport bio-data page. The chip also includes an electronic copy of the bearer's photo. The digital photograph of the individual provides a facial biometric which can be used for automated identification processes by employing facial recognition technology. Most implementations of e-passports by various countries have a single identifier only, the facial biometric. But the chip has sufficient capacity to include extensions such as finger prints and electronic visas if necessary for future applications.

2.1 Operation of E-passport

An e-passport bearer presents his/her document to a border security officer who scans the MRZ information in the e-passport through a MRZ reader and then places the e-passport near a e-passport reader to fetch data from the microchip. The border security officer verifies the content stored in the chip [Passive Authentication (PA) described in Section 2.5]. ICAO also recommends the use of encryption [Basic Access Control (BAC) Section 2.7] so that the communication between the microchip and the e-passport reader is encrypted. Integrity verification of e-passport data is done using either Active Authentication [(AA) Section 2.6] or passive authentication. Both basic access control and active authentication are optional whereas passive authentication is mandatory.

2.2 Notations

1. LDS — Logical Data Structure
2. SO_D — Security Object Descriptor
3. DS — Document Signer
4. C_{DS} — Certificate of Document Signer
5. KPu_{DS}, KPr_{DS} — Public and private keys of Document Signer
6. C_{CSCA} — Certificate of Country Signing Certification Authority
7. KPu_{CSCA}, KPr_{CSCA} — Public and private key of Country Signing Certification Authority
8. $H'_{LDS_{1\ldots16}}$ — Hash values stored in LDS groups $1 \ldots 16$
9. $H_{SO_{D\,1\ldots16}}$ — Hash values for LDS groups $1 \ldots 16$ stored in SO_D
10. $X_{SK}\{Mesg\}$ — Signature generation by X on message $Mesg$
11. $ENC_K\{Mesg\}$, $DEC_K\{Mesg\}$ — Encryption and decrption on message $Mesg$ using key K
12. $MAC_K\{Mesg\}$, $VMAC_K\{Mesg\}$ — Generation and verification of MAC on message $Mesg$ using key K
13. KPu_{AA}, KPr_{AA} — Public and private keys of e-passport
14. $A \| B$ — Concatenation of two message A and B
15. $A \oplus B$ — XOR of two message A and B

2.3 Data Structure

For interoperability, the ICAO's e-passport guideline provides details on how data should be stored in a microchip. The data elements are grouped together as a Data Group [DG] and collectively stored in a Logical Data Structure [LDS]. The ICAO guideline segregates data elements into 19 data groups and the LDS is categorised into three parts:

1. Mandatory - Data defined by the issuing state or organisation, contains the details recorded in the Machine Readable Zone [MRZ], which include, passport number, passport bearer's name, nationality, date of birth, date of expiry, encoded facial biometric image and checksum of individual data elements that are used to derive the session key.
2. Optional - Data defined by the issuing state or organisation, contains optional biometric data for identification like figure prints, iris scan, displayed identification data like digitised signature and any additional personal or document details like contact details, proof of citizenship and endorsements.
3. Optional - Data defined by the receiving state or organisation, contains details for automated border clearance, electronic visas and other travel records.

The data groups from 1 to 16 are defined by the issuing state and are write protected, whereas the data groups for 17 to 19 will be open for write access to authorised receiving states or organisations. Write access is currently not supported, but ICAO plans to implement them in the second generation of e-passports. The LDS is stored in the microchip using the file system as defined in ISO/IEC 7816-4. The dedicated file [DF] in the chip file system hierarchy stores the encryption, MAC keys used in basic access control protocol, and private key of the e-passport bearer that is used in active authentication protocol. The elementary file [EF] in the chip hierarchy will store the security object descriptors [SO_D] and data groups. The SO_D contains the hashes of LDS data elements digitally signed by the issuing organisation [document signer (DS)] and corresponding certificate [C_{DS}]. An important security feature is that data groups are individually hashed and collectively signed by the issuing state and stored in SO_D, thus binding the biometric details with the e-passport bearer details.

The PKI section of the ICAO's e-passport document (ICAO 2005) makes an important distinction between an issuing state and an issuing organisation. The issuing state represents the country of e-passport origin whereas the issuing organisation represents a passport issuing office within a country.

2.4 E-passport PKI

Each $CSCA$ is required to forward their self-signed certificate [C_{CSCA}], document signer certificates [C_{DS}] and certificate revocation lists [CRL] to ICAO to be published at ICAO PKI directory [PKD]. ICAO also recommends that issuing states replicate the PKD and CRL both locally and bilaterally among participating states every 90 days.

ICAO suggests the C_{DS} be also stored in an e-passport chip, so a border security officer could continue with active authentication in case a PKD is unavailable, but this can compromise security as described later in §4.

2.5 Passive Authentication (PA)

Mandatory passive authentication mechanism provides only a basic level of security, as it is still vulnerable to skimming or eavesdropping attacks. Currently U.S.A is the only country that is implementing this level of security. Due to considerable debate and pressure from researchers and privacy advocates, the U.S. government is considering other optional security measures recommended by ICAO. Passive authentication is used to verify the integrity and to authenticate the data stored in the LDS and SO_D, thereby authenticating the e-passport bearer.

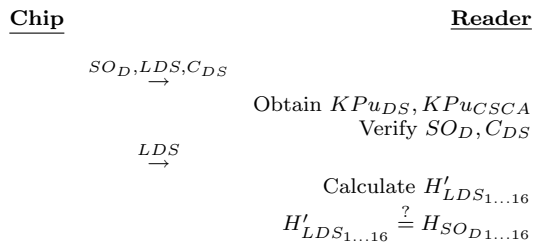

Figure 1: Passive Authentication

2.6 Active Authentication (AA)

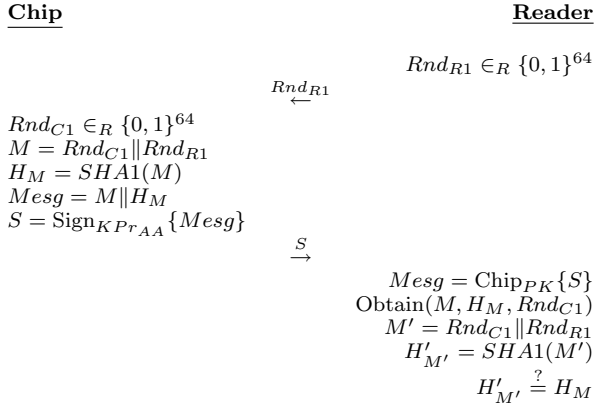

Figure 2: Active Authentication

Active authentication is an optional security feature that relies on public key cryptography to protect against chip modification or chip cloning. The ICAO guideline uses ISO/IEC 7816 *Internal Authenticate mechanism* along with signature computation according to ISO 97986-2 *Digital Signature scheme 1.* The protocol is initiated by the reader by sending a 8 byte random nonce to the e-passport. On receiving a challenge from the reader the chip digitally signs and returns the result. The reader then verifies the signature using KPu_{AA} obtained from SO_D.

2.7 Basic Access Control (BAC)

Basic access control is an optional security mechanism that uses ISO 11770-2 *Key Establishment Mechanism 6* to form a secure communicational channel between a reader and a chip. The protocol uses two secret keys $[K_{ENC}, K_{MAC}]$ that are stored in the e-passport chip. The reader derives both these keys using scannable data present in MRZ, namely e-passport number, date of birth of the e-passport bearer, date of e-passport validity and check digits for those values. The three-pass challenge-response

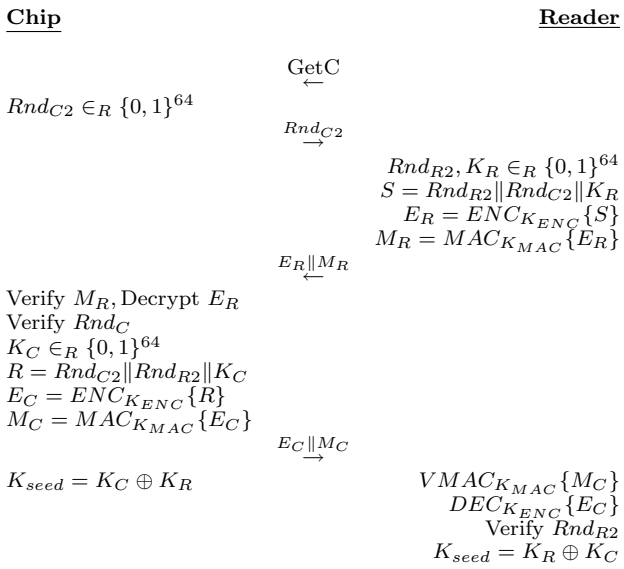

Figure 3: Basic Access Control

protocol is initiated by the reader by requesting a challenge from the chip. On receiving the challenge $[Rnd_{C2}]$, the reader creates a checksum according to ISO/IEC 9797-1 *MAC algorithm 3* over the ciphertext that contains the reader's response to the chip's challenge Rnd_{R2} and the keying material K_R. The chip on obtaining the reader's response creates a checksum that includes its keying material K_C. Both the reader and chip verify the MAC's obtained and decrypt the encrypted message to reveal both keying materials that form the "key seed" K_{seed}. The K_{seed} is then used to derive a shared session key using the key derivation algorithm described in §2.8.

2.8 Key Derivation

The value c is a 32 bit counter that allows for deriving multiple keys from a single seed. Depending on the whether a key is used for encryption or for MAC, a value is assigned to c:

- $c = 1$ (ie., '0x 00 00 00 01') for encryption
- $c = 2$ (ie., '0x 00 00 00 02') for MAC

The following steps are performed to derive both encryption and MAC keys that are to be used in 3DES.

1. $D = K_{seed} \| c$
2. $H_{1...20} = SHA-1(D)$
3. $k_a = H_{1...8}$ and $k_b = H_{9...16}$
4. Adjust parity bits for k_a and k_b to form correct DES keys.

3 Analysis of E-passport

Passports are used as a primary form of identification and because of the nature of contents that is stored [biometric and personal details] within an e-passport's chip, it is crucial that the document is tamper-resistant and also maintains secrecy of data. DOC 9303 (ICAO 2005) provides a brief description of security goals that are achieved and cannot be achieved by the proposed mandatory and optional security mechanisms. If a country implements only the

Method	Security benefits	Vulnerabilities/Weaknesses
Passive Authentication	• Provides authenticity, integrity for SO_D and LDS	• Failure to detect chip substitution. • Failure to prevent against chip copy, unauthorized access and skimming.
Active Authentication	• Prevents against duplication of SO_D and chip modification	• Implementation complexity as extra resources (Memory, CPU) are needed.
Basic Access Control	• Prevents against skimming and eavesdropping	• Failure to detect chip substitution. • Failure to prevent against chip copy. • Implementation complexity as extra resources (Memory, CPU) are needed.

Table 1: DOC9303 security benefits and drawbacks

mandatory security requirement [PA], then authenticity and integrity of both SO_D and LDS are provided. It does not, however, prevent data copy, chip substitution or skimming and also does not prevent against an unauthorised access to e-passport. For a greater security the ICAO recommends the implementation of other security mechanisms like: (1) AA to prevent copying of SO_D and chip substitution and (2) BAC to prevent skimming and eavesdropping on communication between the e-passport chip and the reader. An overview of DOC 9303's security benefits and drawbacks is given in Table 1.

3.1 Security Goals

We analyse e-passport protocols by first identifying their security goals that are both explicit and implicit. We assume that a country implements the highest level of security i.e, a country implements all three security measures namely, PA, AA and BAC.

1. **Data Confidentiality** : Confidentiality is an important requirement to protect secrecy and privacy of e-passport details. Confidentiality also provides protection against forgery and spoofing attacks. To provide data confidentiality, the communication channel between the e-passport reader and the microchip should be secure typically via encryption. An unauthorised party should not have access to any data elements within the LDS or keys stored in the DF of the chip file system that contain session and private keys.

2. **Data Integrity** : A strong integrity mechanism protects against tampering with the chip's contents. The DF, SO_D and LDS should be secure against any unauthorised modifications, i.e., any data tampering should be easily detectable by the border security center.

3. **Data Origin Authentication** : The data on the chip should be bound to information on MRZ and to the data that appears in the e-passport bio-data page currently being examined by a border security officer.

4. **Non Repudiation** : E-passport have an advantage as the e-passport bearer will be physically present at the border security checkpoint. Nevertheless, it would be important to obtain a undeniable digital data from the e-passport for future processing, e.g, in case of an aftermath of a terrorist attack to validate the entry of the e-passport bearer at a particular security checkpoint.

5. **Mutual Authentication** : As described in goal 3, it is important for the e-passport reader to authenticate the e-passport, but it is also important for the e-passport chip to authenticate the e-passport reader before divulging any personal information. This is important to prevent an unauthorised e-passport reader from obtaining biometric and personal details from an e-passport.

6. **Certificate Manipulation** : The e-passport reader should have a guarantee that certificates presented by the e-passport are valid and match the data on the e-passport. ICAO has implemented a PKI (Tom A.F. Kinneging for ICAO-NTWG 2004) which would store signature certificates from issuing state and organisations.

7. **Key Freshness and Key Integrity** : The e-passport reader and e-passport must have satisfactory proof that, nonces generated during both AA and BAC protocols are fresh and the integrity of the derived session key is preserved. Both parties should also have undeniable proof that the other party is in possession of a valid session key. Any previous compromised key should be easily detected and the protocol run should terminate.

8. **Forward Secrecy** : Loss of session key or key used to generate a session key [K_{ENC} and K_{MAC}] should not compromise any future communication.

3.2 Formal Representation

Model checking approach has been very successful in finding faults in many protocols (J. C. Mitchell et al. 1997, Lowe 1996, Lowe & Roascoe 1997, N. Heintze & J. D. Tygar 1994, S. Schneider 1997, Z Dang & R. A Kemmerer 1997). The approach is based on modelling a protocol as a finite state system by specifying its properties and then using a model checker to verify the systems properties. The advantages of using model checkers is that the verification process is usually automated and if a verification fails on a required property the model checker lists the sequence of events that led to the property being broken. Casper (Gavin Lowe 1999) developed by Gavin Lowe, is a compiler which converts a high level specification of the protocol to a CSP (C.A.R Hoare 1985) script. The CSP script can then be run on a model checker like FDR2 (Formal Systems (Europe) Ltd 2003), to verify if the protocol meets its security requirements.

An apparent limitation of this approach is that the verification of a complex protocol suite can lead to a state-space explosion causing the checker to breakdown. Thus a formal model does not cover all aspects of a protocol. Normally the underlying functions are

assumed to be true. The verification of the simplified protocol that was formalised does not necessarily mean the full version of the protocol is secure against attacks but only suggest the protocols requirements are satisfied. Nevertheless it does provide an assurance to users and designers about the relevant security goals that are met by the protocol.

3.3 Modelling E-passport protocols

ICAO e-passport is a complex protocol suite that consists of three protocols, BAC, PA and AA. Such a complex protocol suite are not only difficult to formalise, but also verification of such systems more often leads to state-space explosions. Therefore, we do not find many publicised work on verification of such systems and to the best of our knowledge E-passport protocols have not been formally verified.

We model the flow of e-passport protocol as follows:

1. When an e-passport is presented at a border security checkpoint, the chip and the e-passport reader execute the BAC protocol, in order to establish a session key to secure all future communication between them.

2. On successful completion of BAC, the e-passport reader performs PA.

3. On successful completion of PA the chip and the e-passport reader execute the AA protocol.

The e-passport authentication mechanism heavily relies on PKI. We model only one level of certification hierarchy, up to document signer and we assume that document signer public key certified by its root (country signing authority) is valid and secure. This does not weaken the verification process of e-passport protocol suite, but only indicates that the model does not considers any weaknesses that might exist in PKI implementation by countries and ICAO. We also assume that cryptographic primitives used in the system like hash functions, MAC, and generation of keys (3-DES) are secure against various forms of attacks that exist in literature. Our modelling of e-passport protocols using Casper is presented in Appendix A.

3.4 Interpreting FDR output

FDR2 (Formal Systems (Europe) Ltd 2003) is a model-checking tool for state machines, with foundations in theory of concurrency based around Hoare's Communicating Sequential Processes [CSP] (C.A.R Hoare 1985). The verification technique is based on the method of establishing whether a property holds by testing for refinement of a transition system and the ability to check determinism of a state machine that is primarily used for checking security properties. FDR2 is designed to mechanise the process of carrying out refinement checks.

Casper (Gavin Lowe 1999) generates refinement assertions to check for all specifications. It generates one assertion for all secret specifications and one assertion for each agreement and aliveness specification. A CSP script file includes statements making assertions about refinement properties. These statements will typically have the following form:

`assert Abstract [X= Concrete`

Example: Specification `Secret(B, message, [A])` specifies that, at the end of a protocol run, entity `B` expects the value of `message` to be known *only* to entity `A`. Assertion generated for the above specification is:

```
SECRET_M::SECRET_SPEC[T=SECRET_M::SYSTEM_S
```

The selected assertion is submitted for testing by choosing the *Run* option in FDR2. FDR2 then attempts to prove the conjecture by compiling, normalising, and checking the refinement. If we find a refinement that is not satisfied, then there might be a weakness in the protocol. To examine the weakness, the FDR2 debugger is invoked, allowing the behaviour of the processes involved to be examined. The information displayed depends on the nature of the counterexample being examined and the contribution made to it by the selected component. The weakness in the protocol is examined by observing a trace leading to divergence.

4 Verification Using Casper/FDR

In E-passports data confidentiality is provided by the BAC protocol, whereas the integrity of chip's contents LDS and SO_D is verified by the reader using the PA and AA protocols. The keys K_{ENC} and K_{MAC} are stored in DF on e-passport and are generated by the reader before initiating communication by using the data in MRZ which includes e-passport number, date of birth, e-passport validity date, and corresponding check digits. The ICAO e-passport guideline states that the entropy of the key is at most 56 bits. The Juels *et al.* (A Juels et al. 2005) analysis of U.S passports reports that the entropy of the key can further be reduced to ≈ 52 bits because of U.S. e-passport numbering scheme, as first two digits are assigned to 15 e-passport issuing offices. Because of a low entropy, the key would be vulnerable to brute force attacks as demonstrated by (Adam Laurie 2007).

Analysis of the e-passport protocol using the Casper and FDR2 verification software proves that the protocol is vulnerable to the Grandmaster Chess Attack (Yvo Desmedt et al. 1987). Compiling with security specifications creates corresponding refinement assertions.

The secrecy specification results in an assertion `SECRET_M::SECRET_SPEC [T= SECRET_M::SYSTEM_S` and its verification using FDR2 results in an erroneous trace after 30 states with 135 transitions and FDR2 debugger reveals:

```
send.Reader.Chip(Msg1.GETC,<>)
INTRUDER_M::say.GETC
send.Chip.Reader.(MSG2,RNDC2,<>)
INTRUDER_M::say.RNDC2
send.Reader.Chip(Msg3,Sq.<
        Encrypt.(KEYE,<RNDR2,RNDC2,KR>),
        Encrypt(KeyM,<RNDR2,RNDC2,KR>)>)
INTRUDER_M::say.Sq<
        Encrypt.(KEYE,<RNDR2,RNDC2,KR>),
        Encrypt(KEYM,<RNDR2,RNDC2,KR>)>
```

which can be intepreted as:

```
1. Reader -> I_Chip : GETC
1a. I_Chip -> Chip : GETC
2. Chip -> I_Chip : {RNDC2}
2a. I_Chip -> Reader : {RNDC2}
3. Reader -> I_Chip :
        {RNDR2, RNDC2, KR}{KEYE},
        {RNDR2, RNDC2, KR}{KEYM}
3a. I_Chip -> Chip :
        {RNDR2, RNDC2, KR}{KEYE},
        {RNDR2, RNDC2, KR}{KEYM}
4. Chip -> I_Chip :
        {RNDR2, RNDC2, KC}{KEYE},
        {RNDR2, RNDC2, KC}{KEYM}
4. I_Chip -> Reader :
        {RNDR2, RNDC2, KC}{KEYE},
        {RNDR2, RNDC2, KC}{KEYM}
```

and for assertion AUTH1_M::AuthenticateRESPONDERTo INITIATORAliveness [T= which corresponds to the security goal that an e-passport believes that it is involved in a conversation with the reader. Its verifications using FDR2 results in an erroneous trace after 12 states with 35 transitions and FDR2 debugger reveals:

```
send.Reader.Chip.(Msg1,GETC,<>)
INTRUDER_M::hear.GETC
send.Reader.Chip.(Msg3,Sq.<
        Encrypt.(KEYE,<RNDR2,KM,KR>),
        Encrypt.(KEYM,<RNDR2,KM,KR>)>,<>)
INTRUDER_M::hear.Sq.<
        Encrypt.(KEYE,<RNDR2,KM,KR>),
        Encrypt.(KEYM,<RNDR2,KM,KR>)>
INTRUDER_M::say.Sq.<
        Encrypt.(KEYE,<RNDR2,KM,KR>),
        Encrypt.(KEYM,<RNDR2,KM,KR>)>
```

which can be interpreted as:

```
1. Reader -> I_Chip : GETC
2. I_Chip -> Reader : KM
3. Reader -> I_Chip :
        {RNDR2, KM, KR}{KEYE},
        {RNDR2, KM, KR}{KEYM}
4. I_Chip -> Reader :
        {RNDR2, KM, KR}{KEYE},
        {RNDR2, KM, KR}{KEYM}
```

The trace from the security assertion can be interpreted as, an intruder during communication with a reader is basically replaying messages from the chip, i.e, the reader establishes a session key even though it is not sure if a chip is genuine.

Can this weakness be exploited? Once a secure communication is established between reader and chip, the reader retrieves data stored within the LDS and performs an integrity verification using issuing state's certificate. A border security officer on receiving evidence that LDS has not been tampered with would authenticate an e-passport bearer by using the facial biometric image stored in LDS against the person physically present at the checkpoint. Therefore even if the messages are only being replayed the data still has to be from an issuing state certified chip. This weakness can be exploited as facial biometrics is view-dependent and are prone to inter-class similarities within large population groups such as identical twins, similar ethnic groups and certainly possible in case of human cloning. As the probability of uniqueness using facial biometric is low, it is certainly possible that a border security officer might not be able to differentiate between the facial biometric data in the LDS and the person physically present at the checkpoint. The authors in (P. J. Phillips et al. 2000) also pointed out that the false rejection rate could be as high as 43% as majority of algorithms used in facial biometrics are subject to illumination issues and also depend on the type of camera used to obtain the initial image. Note that e-passports store high-resolution images of the e-passport bearer to make verification independent on the processing algorithms used by various countries. This introduces another serious security weakness - an attacker can manipulate less significant bits of images to find collisions for the hash functions used.

Even with these drawbacks, BAC is important and should be implemented as it prevents against eavesdropping. The protocol is vulnerable to replay attacks but an intruder cannot decrypt values [E_C or E_R] used to form the session key [K_{seed}].

The AA protocol in addition to providing integrity also protects the e-passport against chip modification i.e, it binds LDS with the e-passport bearer's secret key $Chip_{SK}$ and authenticates the e-passport microchip. Our verification of an ideal AA protocol i.e., assuming that the BAC protocol was carried out in a secure way, indicates that there is no security weakness in the protocol.

Assertions

```
SECRET_M::SECRET_SPEC [T= SECRET_M::SYSTEM_S
AUTH1_M::AuthenticateRESPONDERToINITIATOR
        Aliveness [T= AUTH1_M::SYSTEM_1
AUTH2_M::AuthenticateINITIATORToRESPONDER
        Aliveness [T= AUTH2_M::SYSTEM_2
AUTH3_M::AuthenticateINITIATORToRESPONDER
        Agreement_rndr1 [T= AUTH3_M::SYSTEM_3
AUTH4_M::AuthenticateRESPONDERToINITIATOR
        Agreement_rndc1 [T= AUTH4_M::SYSTEM_4
```

which corresponds to secrecy, authentication of an e-passport to reader and from reader to an e-passport does not yield any erroneous traces. But if we consider that an intruder was able to successfully run the BAC protocol with the reader by obtaining K_{ENC} and K_{MAC} by performing a brute force attack as in (Adam Laurie 2007) and thus successfully able to compute session key K_{seed}, then assertions:

```
SECRET_M::SECRET_SPEC [T= SECRET_M::SYSTEM_S
AUTH2_M::AuthenticateINITIATORToRESPONDER
        Aliveness [T= AUTH2_M::SYSTEM_2
AUTH3_M::AuthenticateINITIATORToRESPONDER
        Agreement_rndr1 [T= AUTH3_M::SYSTEM_3
```

yields erroneous traces which indicates that weakness exists in the protocol.

Assertion SECRET_M::SECRET_SPEC [T= SECRET_M:: SYSTEM_S yields an error trace after 4 states and 8 transitions and analysis using the FDR2 debugger revels the following first level trace.

```
send.Reader.Chip.(Msg1,Encrypt.
        (KEYCR,<RNDR1>),<RNDR1>)
leak.RNDR1
```

This attack is obviously true, as the intruder is now in possession of the session key and therefore able to decrypt any communication between the chip and the reader. This would compromise the privacy of an e-passport bearer as his/her personal details would be compromised and increase the risk of identity fraud.

Assertion AUTH3_M::AuthenticateINITIATORTo RESPONDERAgreement_rndr1[T=AUTH3_M::SYSTEM_3 yields an erroneous trace after 8 states and 149 transitions. FDR2 debugger revels the following second level trace

```
env.Chip.(Env0,Reader,<RNDC1,Reader>)
receive.Reader.Chip.(Msg1,
        Encrypt.(KEYCR,<RNDM1>),<RNDM1>)
signal.Commit3.
        RESPONDER_role.Chip.Reader.RNDM1
```

From the above traces we can interpret that an attacker is able to successfully authenticate to the reader as a genuine e-passport. This is possible because the session key is compromised. This attack is theoretically possible but practically would not be easy to implement, as the data is protected by digital signature and is computationally impossible to generate a valid signature for a modified data. Nevertheless this weakness can be exploited by the attacker in lieu with weakness in facial biometric systems as discussed above. The combination of weakness in both BAC and AA can be exploited by the intruder. An attacker can now make a copy of the e-passport and authenticate successfully, defeating the primary security goals of BAC and AA, to prevent against chip substitution and chip copy.

Assertion `AUTH2_M::AuthenticateINITIATORToRESPONDERAliveness[T=AUTH2_M::SYSTEM_2` yields an error trace after 3 state and 6 transitions and the FDR2 debugger revels the following second level trace

```
env.Chip.(Env0,Reader,<RNDC1,Reader>)
receive.Reader.Chip.(Msg1,Encrypt.
        (KEYCR,<RNDM1>),<RNDM1>)
signal.Commit2.RESPONDER_role.Chip.Reader
```

The above traces points to an important security goal that is not met: mutual authentication between a chip and a reader. The reader believes that it has successfully authenticated the chip but, there is no proof that the chip has successfully authenticated the reader. Authentication of reader by the chip depends on the fact that only a genuine reader would be able to obtain K_{ENC} and K_{MAC} from MRZ to perform BAC protocol and compute the session key K_{seed} used in AA protocol. We have seen that it is not necessarily true. An attacker who is in possession of the keys K_{ENC} and K_{MAC} [because of low entropy and brute force attacks as in (Adam Laurie 2007)] will be able to masquerade as a reader and successfully authenticate itself to the chip.

From the above traces it is also clear that the e-passport protocol does not satisfy any key related security goals like freshness and integrity. Key integrity is not satisfied as an attacker is able to successfully run the BAC protocol and obtain the session key K_{seed} used to form a secure communication channel. There are no guarantees provided to either the chip or the reader regarding key freshness. The nonces generated by either reader, chip or both may not contain enough randomness that is necessary for a security protocol. An eavesdropper might be able to collect information about several runs of the protocol and perform a cipher-text with known partial plain-text attack to obtain the session key and/or MRZ information that is necessary to create K_{ENC} and K_{MAC}. This would also compromise the security goal of forward secrecy. An e-passport has an average validity of around 10 years. Any loss of K_{ENC} or K_{MAC} keys make the e-passport vulnerable to skimming and snooping attacks.

We were unable to make an formal analysis of security goals non-repudiation and certificate manipulation, but an informal analysis of e-passport protocols suite reveals they may also be prone to infrastructure based attacks. Public key certificates [for both document signer and country signing certificates] are held by ICAO in a central repository. The ICAO's e-passport guideline states that each border security checkpoint should update their certification hierarchy list individually. This is necessary to perform a valid verification during the AA protocol, as the secret key of an e-passport is certified by the issuing country. The drawback is that a attacker may be able to mount a DOS attack on the border security checkpoint certificate server before arriving or in co-ordination with others to prevent the certificate server from updating and thus preventing the border security checkpoint from verifying validity of e-passport signature, as the border security checkpoint now relies on C_{DS} that is stored in the chip and will not have an updated revocation list. ICAO e-passport guideline acknowledges this issue and states that in such a case a border security checking officer should rely on conventional method that were in place before e-passport for verification of the e-passport bearer. But this defeats the entire purpose of introducing e-passports.

5 Conclusion

Formal methods have become an integral part in verification of protocols. We have used the Casper and FDR model checker to verify security of Australian e-passport implementation that is based on ICAO e-passport protocol suite and our analysis have shown that current security measures that are in place are weak. A primary weakness is that the data security techniques deployed in current generation of e-passports does not adequately protect an e-passport bearer as keys have a very low entropy and are vulnerable to brute force attacks.

Our formal analysis shows that ICAO e-passport guideline does not meet our security goals.

- The e-passport protocols does not satisfy our goal for data origin authentication as it can be subject to replay and grandmaster chess attacks, and the weakness can be exploited in cases where problems with facial biometric exists.
- Data confidentiality is also compromised when an attacker is able to obtain encryption and MAC keys stored in the e-passport chip using information presented in MRZ.
- We were able to prove that this further affects the security goals for active authentication protocol, namely, mutual authentication, key freshness and key integrity.
- An informal analysis of the e-passport system reveals that it may also be vulnerable to certificate manipulation as they are dependent on PKI, which is prone to DOS attacks.

Electronic passport are an important step in the right direction. It enables countries to digitise their security at border control and provides faster and safer processing of an e-passport bearer. E-passports introduces facial biometric recognition for verification of an e-passport bearer, which is less intrusive when compared with other biometric systems. But facial biometric are not very secure because of relatively low uniqueness and are prone to inter-class similarities.

The risk of identity theft or illegal entries into a country are further increased when e-passports can be used as in (Service 2006), that are currently on trial in Australia. Unattended border control check-ins increase the risk of fraudulent facial biometric verifications being undetected and eavesdropping on communication between e-passport and reader.

References

A Juels, D Molnar & D Wagner (2005), Security and privacy issues in e-passports, *in* 'IEEE SecureComm '05'.

Adam Laurie (2007), 'Rfidiot', http://rfidiot.org/.

Bobbie Johnson (2006), 'Hackers crack new biometric passports', The Guardian.

C.A.R Hoare (1985), *Communicating Sequential Processes*, Prentice Hall International.

David Reid (2006), 'epassports 'at risk' from cloning', BBC.

Formal Systems (Europe) Ltd (2003), *Failuers-Divergence Refinement, FDR2 User Manual*. Available from http://www.fsel.com/.

Gaurav S. Kc & Paul A. Karger (2005), 'Preventing attacks on machine readable travel documents (mrtds)', Cryptology ePrint Archive, Report 2005/404. http://eprint.iacr.org/.

Gavin Lowe (1999), *Casper - A compiler for the analysis of security protocols, User Manual and Tutorial, Ver1.3.*

ICAO (2005), Machine readable travel documents, Part-1, Machine Readable Passport Volume 2, Specifications for Electronically Enabled Passports with Biometric Identification Capability, ICAO.

ISO/IEC (2000), 'ISO/IEC14443, identification cards – contactless integrated circuit(s) cards – proximity cards'.

J. C. Mitchell, M. Mitchell & U. Stern (1997), Automated analysis of cryptographic protocols using murphi, *in* '16th IEEE Symposium on Security and Privacy', IEEE Computer Society Press.

Lowe, G. (1996), Breaking and fixing the needham-schroeder public-key protocol using csp and fdr, *in* T. Margaria & B. Steffen, eds, 'Tools and Alogrithms for the Construction and Analysis of Systems', Vol. 1055 of *LNCS*, Springer-Verlag, pp. 147–166.

Lowe, G. & Roascoe, B. (1997), Using csp to detect errors in the tmn protocols, *in* 'IEEE Transactions on Software Engineering', Vol. 3.

N. Heintze & J. D. Tygar (1994), A model for secure protocols and their compositions, *in* '1994 IEEE Computer Society Symposium on Research in Security and Privacy', IEEE Computer Society Press, pp. 2–13.

P. J. Phillips, A. Martin, C. L. Wilson & M. Przybocki (2000), 'An introduction evaluating biometric systems', *IEEE Computer* **33**(2), 56–63.

S. Schneider (1997), Verifying authentication protocols with csp, *in* '10th IEEE Computer Security Foundations Workshop', IEEE Computer Society Press, pp. 2–17.

Service, A. C. (2006), 'Smartgate', http://www.customs.gov.au/site/page.cfm?u=5555.

Tom A.F. Kinneging for ICAO-NTWG, P. T. F. (2004), PKI for machine readable travel documents offering ICC read-only access, Technical report, ICAO. Version 1.1.

Will Knight (2006), 'Hackers clone radio-chip passports', NewScientist.

Yvo Desmedt, Claude Goutier & Samy Bengio (1987), Special uses and abuses of the fiat-shamir passport protocol, *in* 'Advances in Cryptology - CRYPTO '87', Vol. 293, Springer Berlin / Heidelberg, pp. 21–39.

Z Dang & R. A Kemmerer (1997), Using the astral model checker for cryptographic protocols analysis, *in* H. Orman & C. Meadows, eds, 'Workshop on Design and Formal Verification of Security Protocols'.

Appendix A Casper Representation

The Casper script provided below, presents a combined representation of all three protocols and does not consider modification that are need when verifying security properties for individual protocols.

```
#Free variables
C,R,DS : Agent
getc : InitializeConv
lds : DataGroups
sod : SecurityObject
rndr2,rndc2,kr,kc,rndr1,rndc1 : Nonce
h : HashFunction
PK : Agent -> PublicKey
SK : Agent -> SecretKey
keyM,keyE,keyCR : SessionKey
InverseKeys = (PK,SK), (keyM,keyM), (keyE,keyE),
        (keyCR,keyCR)

#Processes
INITIATOR(R,C,getc,rndr1,rndr2,kr,keyM,keyE,keyCR)
        knows PK,SK(R)
RESPONDER(C,R,rndc1,rndc2,kc,keyM,keyE,keyCR)
        knows PK,SK(C)

#Protocol description

0.  -> C   : R
0a. DS -> C : {C,PK(C)}{SK(DS)} % CERTC
0b. DS -> R : {C,PK(C)}{SK(DS)}
1. R -> C  : getc
2. C -> R  : rndc2
3. R -> C  : {rndr2,rndc2,kr}{keyE},
        {rndr2,rndc2,kr}{keyM}
4. C -> R  : {rndr2,rndc2,kc}{keyE},
        {rndr2,rndc2,kc}{keyM}
---
5. C - > R : {LDS,SOD}{KeyCR},
        {C,PK(C)}{SK(DS)} % CERTC
---
6. R -> C  : {rndr1}{keyCR}
7. C -> R  : { {h(rndc1,rndr1), rndr1,rndc1}
        {SK(C)} }{keyCR}


#Specification
StrongSecret(C,kr,[R])
StrongSecret(C,kc,[R])
StrongSecret(R,kr,[C])
StrongSecret(R,kc,[C])
Aliveness(C,R)
Aliveness(R,C)
Agreement(C,R,[kr,kc])
StrongSecret(C,rndr1,[R])

#Actual variables
Chip,Reader,DSigner,Mallory : Agent
GETC : InitializeConv
LDS : DataGroups
SOD : SecurityObject
RNDR2,RNDC2,RNDM2,KR,KC,KM,RNDR1,RNDC1 : Nonce
KEYM,KEYE,KEYCR, KEYMM,KEYEM : SessionKey
InverseKeys = (KEYM,KEYM), (KEYE,KEYE),
        (KEYMM,KEYMM), (KEYEM,KEYEM), (KEYCR,KEYCR)

#Functions

symbolic PK,SK

#System
INITIATOR(Reader,Chip,GETC,RNDR1,RNDR2,KR,
        KEYM,KEYE,KEYCR)
RESPONDER(Chip,Reader,RNDC1,RNDC2,KC,
        KEYM,KEYE,KEYCR)
CERTAUTH(DS,C,R) knows PK,SK(DS)

#Intruder Information

Intruder = Mallory
IntruderKnowledge = {Chip,Reader,RNDM2,KM,PK,
        SK(Mallory),KEYMM,KEYEM}
```