E-Passport: Cracking Basic Access Control Keys*

Yifei Liu, Timo Kasper, Kerstin Lemke-Rust, and Christof Paar

Horst Görtz Institute for IT Security
Ruhr University Bochum
Germany
{yliu,tkasper,lemke,cpaar}@crypto.rub.de

Abstract. Since the introduction of the Machine Readable Travel Document (MRTD) that is also known as e-passport for human identification at border control debates have been raised about security and privacy concerns. In this paper, we present the first hardware implementation for cracking Basic Access Control (BAC) keys of the e-passport issuing schemes in Germany and the Netherlands. Our implementation was designed for the reprogrammable key search machine COPACOBANA and achieves a key search speed of 2^{28} BAC keys per second. This is a speed-up factor of more than 200 if compared to previous results and allows for a runtime in the order of seconds in realistic scenarios.

Keywords: E-Passport, MRTD, Basic Access Control, Key Search Machine, SHA-1, DES, COPACOBANA.

1 Introduction

The United States and several other countries are engaged in the development of a new border control system that is based on biometric identification and RFID (Radio-Frequency Identification) technologies. Specifications for MRTDs (Machine Readable Travel Documents) that are also known as e-passports are issued by the ICAO (International Civil Aviation Organization) [29,28,25,26,24,27]. Some states, e.g., Germany, the Netherlands, and Belgium already started issuing electronic passports. For the storage of biometric data an IC (Integrated Circuit) with an RF (Radio Frequency) interface is embedded in the passport document.

Public debates on security and privacy issues have been raised on the use of RFID and biometric technology in various applications. A valuable overview on security and privacy threats in e-passports is provided in [20]. Related work on e-passports can also be found in [21,17]. Promoters of the MRTD system promise that by using 'machine readable visas and/or passports as a source of reliable data, governments can build useful data bases that can serve as a uniform source of information in standardized format to speed the border control process' [5]. Further benefits are said to lie in 'the creation of data bases shared voluntarily,

* Supported by the European Commission through the IST Contract IST-2002-507932 ECRYPT, the European Network of Excellence in Cryptology.

R. Meersman and Z. Tari et al. (Eds.): OTM 2007, Part II, LNCS 4804, pp. 1531–1547, 2007.

even across national boundaries, and between the public and private sectors. This will make it easier to identify people who are traveling with stolen documents, and people who have fraudulently obtained an otherwise valid passport based upon stolen citizenship document forms.' [5].

This contribution concentrates on the Basic Access Control (BAC) that establishes a secured channel between the RFID reader that is part of the inspection system and the e-passport for providing both confidentiality and integrity of the data communication. BAC deploys symmetric cryptography and generates the corresponding encryption and authentication keys from passport identification numbers that are visible in the physical passport document and is, e.g., implemented in Germany, the Netherlands, and Belgium. The scheme has already been compromised using offline dictionary attacks in the Netherlands, where experiments demonstrated that the encrypted information can be revealed in three hours after intercepting the communication [10,30] because of weaknesses in the passport numbering scheme. Similar flaws in the passport issuing schemes have been reported for Germany [13] and Belgium [11].

Cryptanalytical tools such as brute-force machines examine the soundness of security claims for cryptographic solutions and hence yield figures about real efforts needed for practical cryptanalysis. This knowledge may help in assessing and possibly avoiding privacy and security risks that are imposed on the individual. With this background in mind, we feel that there is a public interest in determining of how efficient key search algorithms on the BAC keys can be mounted in practice. In this contribution we concentrate on the practical use of special purpose hardware. Therefore, we designed and implemented a hardware architecture for the FPGA based machine COPACOBANA (Cost-Optimized Parallel Code Breaker) [22].

This paper is organized as follows. In Section 2 we explain the BAC protocol and the key derivation scheme. The underlying threat model for our attack is given in Section 3, for which Section 4 provides concrete adversaries and settings to form applicable scenarios for the key search. Details about the practical implementation and results are given in Section 5, and Section 6 considers further directions.

2 The Basic Access Control Protocol (BAC)

Personalization of an e-passport includes printing an MRZ (Machine Readable Zone) on the paper document that can be optically scanned by an inspection system at the border control. As illustrated in Fig. 1, the MRZ consists of two lines containing amongst others personal data such as name, sex, date of birth, and the nationality of the owner. The particulars of the second line are of special importance for the e-passport as they are used for the derivation of the BAC keys. The necessary fields are

- the passport number (9 alphanumeric characters),
- the date of birth of the passport holder (6 characters), and
- the date of expiry of the passport (6 characters).

Each field additionally includes a numeric check digit.

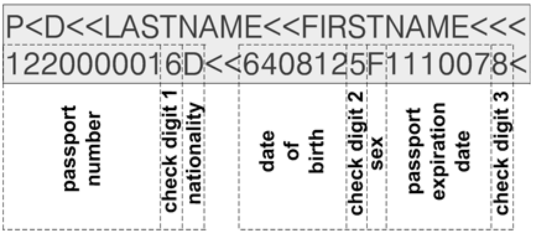

Fig. 1. An Exemplary MRZ of the German E-Passport

Before any personal information can be read from an e-passport via an RFID reader, the BAC protocol needs to be carried out. In case of a successful mutual authentication, the parties agree on a session key that is used for the encryption of the subsequent exchange of information[1].

As illustrated in Fig. 2, first K_{Seed} is derived as the most significant 16 bytes by applying the SHA-1 [7] to the MRZ information. From K_{Seed} both an encryption key K_{ENC} and a key K_{MAC} for the Message Authentication Code (MAC) are obtained. For their key derivation, two different constants are used: C_0 =‘00 00 00 01’ for K_{ENC} and C_1 =‘00 00 00 02’ for K_{MAC}. The most significant 16 bytes of the SHA-1 computation form the Triple-DES [8] keys of K_{ENC} and K_{MAC}, respectively.

Based on the access keys K_{ENC} and K_{MAC}, session keys are established using a three-pass authentication protocol with random numbers. The protocol runs between the RF reader that is part of the inspection system and the MRTD chip as shown in Fig. 3 (see also [20,26]).

As result of Fig. 3, the session key KS_{Seed} is computed as $KS_{Seed} = K_{IFD} \oplus K_{ICC}$. The Triple-DES session keys KS_{ENC} and KS_{MAC} are obtained from KS_{Seed} by applying the same key derivation scheme as depicted in Fig. 2 for K_{ENC} and K_{MAC}. The subsequent communication transfers personal data records from the e-passport and is secured with KS_{ENC} and KS_{MAC}.

3 The Threat Model

Our threat model was initially introduced in [13] and is illustrated in Fig. 4. We propose a hardware architecture that consists of two parts: The front-end is an RF eavesdropper that can continuously read and record RF based communication at public places with a high e-passport density, e.g., nearby inspection systems at airports. Optionally, a surveillance camera may take pictures of the particular passport holder. The back-end is a cryptanalytic system that is connected to databases as well as to hardware or software modules for fast cryptanalysis of symmetric ciphers. It consists of, e.g., the reprogrammable machine

[1] Note that a reading access to more sensitive data like digital fingerprints and iris scans may require a further authentication mechanism, e.g., in Germany the Extended Access Control (EAC) [2].

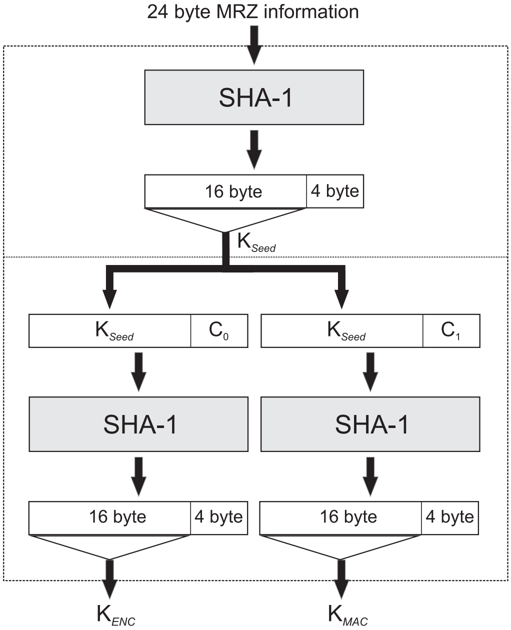

Fig. 2. Basic Access Key Derivation

COPACOBANA (Cost-Optimized Parallel Code Breaker), which is optimized for running cryptanalytical algorithms [22,23]. When BAC keys are compromised the revealed personal information such as name, sex, date of birth, nationality, passport number, date of expiry, and a facial image of the passport holder are inserted into databases. Once stored in such a database, key search can be applied much more efficiently, e.g., directly based on table entries.

Information in such databases is exploitable by criminals like terrorists or by detectives, data mining agencies, etc. , especially as the correctness of the private data is proven by a certificate of the issuing country and the digital photograph stored in the passport is optimized for automatic face recognition [19]. Ari Juels et al. [20] point out problems that are imposed on e-passport holders such as identity theft, tracking, and hotlisting. In the worst case scenario, an attacker may devise an RFID enabled bomb that is keyed to explode when reading a particular individual's RF identifier [20]. The success of a BAC protocol that is initiated by a criminals' skimming device may be used as such a triggering event. Also, a distant eavesdropper being able to only intercept the data sent from the RF reader to the MRTD can identify a particular e-passport, following the approach detailed in Section 4.2.

For the RFID-communication two different channels are used:

- RFID reader to e-passport (forward channel): This channel supplies the e-passport with energy and is used for transferring data from the reader to the e-passport.

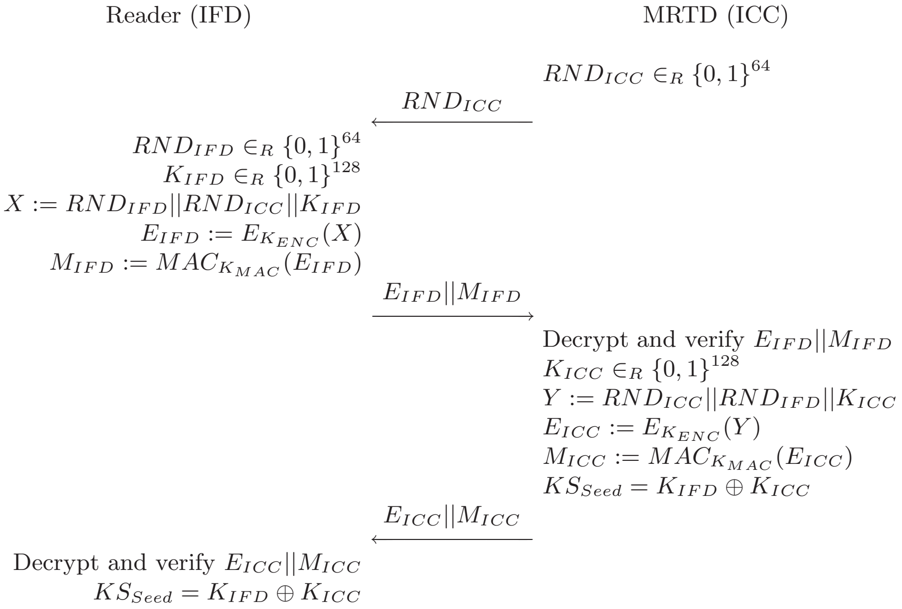
Reader (IFD)
MRTD (ICC)
$RND_{ICC} \in_R \{0,1\}^{64}$
RND_{ICC}
$RND_{IFD} \in_R \{0,1\}^{64}$
$K_{IFD} \in_R \{0,1\}^{128}$
$X := RND_{IFD}||RND_{ICC}||K_{IFD}$
$E_{IFD} := E_{K_{ENC}}(X)$
$M_{IFD} := MAC_{K_{MAC}}(E_{IFD})$
$E_{IFD}||M_{IFD}$
Decrypt and verify $E_{IFD}||M_{IFD}$
$K_{ICC} \in_R \{0,1\}^{128}$
$Y := RND_{ICC}||RND_{IFD}||K_{ICC}$
$E_{ICC} := E_{K_{ENC}}(Y)$
$M_{ICC} := MAC_{K_{MAC}}(E_{ICC})$
$KS_{Seed} = K_{IFD} \oplus K_{ICC}$
$E_{ICC}||M_{ICC}$
Decrypt and verify $E_{ICC}||M_{ICC}$
$KS_{Seed} = K_{IFD} \oplus K_{ICC}$

Fig. 3. Basic Access Control Protocol between the RF reader (also referred to as Interface Device IFD) and the MRTD chip (also referred to as Integrated Circuit Card ICC). E denotes Triple-DES encryption, MAC denotes the cryptographic checksum according to the ISO/IEC 9797-1 MAC Algorithm 3 [26].

- E-passport to RFID reader (backward channel): This channel is used by the e-passport to send its data to the reader.

The signal from the reader to the e-passport is about 80 dB stronger [15] than the so-called load modulation signal which is used for communication on the backward channel, in accordance with the ISO 14443 international standard [18]. Therefore, from an enlarged distance, it is significantly more difficult to observe data on the backward channel than on the forward channel.

However, eavesdropping the two-channel RF communication from several metres poses a real threat, e.g., a recent work by Hancke [16] practically demonstrated that the two-way communication between an RFID reader and an RFID tag can be intercepted from 4 metres. Further, the author states that it is very feasible that this distance can be increased, e.g., with application specific antennas and more complex signal processing. In a concrete setting a far-distance eavesdropper may only be able to monitor the forward channel which is said to be possible from a distance up to about 25 metres [30]. As shown in Section 4 this setting is also sufficient for attacking BAC keys.

This paper focuses on practical realizations of the back-end, specially the cryptanalytic system. We provide implementation results for an efficient key

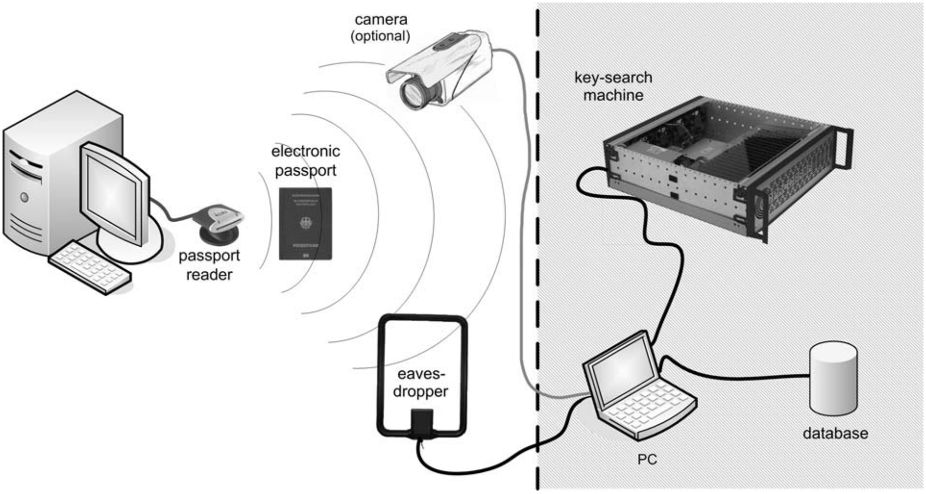

Fig. 4. Architecture of the Attack System

search using the COPACOBANA. Thereby we act on the assumption that the adversary can mount the eavesdropping device in the vicinity of inspection systems.

4 The Key Search

As indicated, two different approaches can lead to success in determining the BAC keys. However, the data records of an e-passport can only be retrieved following the first approach, while the second approach is adequate to gain BAC keys and thus identify a certain passport from a great distance.

4.1 The First Approach Based on Two-Channel Communication

After eavesdropping $RND_{ICC}, E_{IFD}||M_{IFD}$ and $E_{ICC}||M_{ICC}$ of Fig. 3 and the entire subsequent secured communication C the adversary runs a key search on the MRZ information to find a match to the most significant eight bytes of E_{ICC} (see Fig. 3) during the protocol run. More concretely, the adversary computes $E^* = E_K(RND_{ICC})$ where K denotes possible candidates for K_{ENC} and E denotes Triple-DES encryption. If

$$\mathrm{msb}_8(E_{ICC}) \stackrel{?}{=} E^*$$

C can be decrypted and the data records of the e-passport are revealed. For each key candidate, this key search requires two computations of SHA-1 for the key derivation of K_{ENC} and one computation of Triple-DES. However, if one can use pre-computation for the key search, key derivation can be once done beforehand, thus saving two computations of SHA-1 at key search time. The amount of data to be sent to the cryptanalytic module for performing the key search is 16 bytes.

4.2 The Second Approach Based on Forward-Channel Communication

There is an alternative way of discovering the BAC keys if a far-distance adversary does not succeed in eavesdropping the backward channel from the e-passport to the RFID reader. Eavesdropping $E_{IFD}||M_{IFD}$ on the forward channel can be still used for cracking BAC keys by checking

$$MAC_K(E_{IFD}) \overset{?}{=} M_{IFD}$$

where K is a key candidate for K_{MAC}. The knowledge of the MAC key can be exploited for identifying a previously gathered e-passport from the database. Furthermore, if the adversary would get a chance to get closer to an MRTD whose keys are already figured out, it could be activated and read out with a skimming device.

For each key candidate, key search requires two computations of SHA-1 for the key derivation of K_{MAC} from the MRZ information. Further, for the computation of the retail MAC with K_{MAC} according to ISO/IEC 9797-1 one needs to perform four single DES (as E_{IFD} is a ciphertext of 32 byte size) and one Triple-DES for the last padded block. In terms of brute-force this approach requires four additional single DES if compared to the one in Section 4.1. Another drawback for a far-distance adversary is that neither the established session keys nor the transferred data records on the backward channel can be revealed. Accordingly to Section 4.1, if pre-computation is applicable this saves two computations of SHA-1 during the key search. For the second approach, the amount of data to be sent to the cryptanalytic module for performing the key search adds up to 40 bytes.

4.3 Complexity Analysis of the Key Space

The complexity of the key space for BAC keys depends on the passport number issuing scheme that is under control of the issuing state. In this contribution we focus on two issuing states of e-passports: Germany and the Netherlands. The information in Table 1 comes from [30] for the Netherlands and from [9,3,4,6,1] for Germany[2].

The main flaw in the present passport numbering schemes is the low entropy of BAC keys. Low entropy is caused by

1. downsizing the key space of the passport number, i.e., instead of using nine alphanumeric characters for the passport number, mainly numeric characters are used, some of which are even fixed or a check digit,
2. stochastic dependencies between the passport number and the expiry date, e.g., the passport numbers are assigned serially, and
3. dependancy of the key space on publicly available personal data, particularly the date of birth of the passport holder.

[2] There are changes pending on the passport numbering scheme in both states. However, our complexity analysis remains valid for e-passports that are already issued.

Table 1. Special Parameters for Issuing Passports in Germany and the Netherlands

Issuing State:	Germany	The Netherlands
Start of the System:	November 1, 2005	August 26, 2006
Validity of an E-Passport:	10 years	5 years
Passport Numbering:	4 numeric digits for local authority (BKZ) and a serial number of 5 numeric digits, e.g., for Berlin-Mitte with BKZ No. '2598': '259812345'	1 fixed character '**N**' and a serial number of 1 alphanumeric digit and 6 numerical digits followed by a 1 digit checksum, e.g., '**N**F385805**3**'
No. of known BKZs[3]	295	
Individuals owning passports:	approx. 20 Millions	approx. 9 Millions
Issued passports per Working Day:	approx. 8000, i.e., $N_{day}^{G} = 8000$	approx. 7000, i.e., $N_{day}^{NL} = 7000$
Working Days until June 1, 2007:	$T_{June1,2007}^{G} \approx 365 \times 5/7 \times 19/12$, i.e., $T_{June1,2007}^{G} \approx 413$	$T_{June1,2007}^{NL} \approx 365 \times 5/7 \times 9/12$, i.e., $T_{June1,2007}^{NL} \approx 196$

The complexity of the key search strongly depends on assumptions on the adversary's capabilities. We consider three different adversaries $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ as specified in Table 2. The transitions among them may be blurred as acquiring additional BAC keys as result of a successful key search improves the knowledge on issued passports and thereby the configuration of key search algorithms in terms of efficiency. Adversary $\mathcal{A}_1$ with the lowest capabilities knows the public parameters of the e-passport issuing system (see Table 1) but does not know any passport numbers. $\mathcal{A}_2$ already owns a sparely filled database of BAC keys that may be gained by collecting passport data from customers, e.g., at hotels or car rental companies. This previous knowledge allows $\mathcal{A}_2$ to predict the stochastic dependency between the passport number and the expiry date for the issuing state. $\mathcal{A}_3$ is the adversary achieving maximum power. It has access to a complete database with BAC keys, e.g., as a result of social engineering attacks inside the infrastructure of the e-passport system or by participating in databases shared by public and private sectors.

Another important factor for cryptanalysis is the amount of information that is available as a result of eavesdropping during a BAC protocol instantiation. Here, we distinguish five settings (see Table 3). For all settings we assume that the issuing state of the passports is known, e.g., by observing special protocol information in the ATS (answer to select) response of the e-passport. Setting S_1

[3] Note that the coverage of known BKZs among all BKZs in Germany is not publicly available. The number of known BKZs stems from [4].

Table 2. Capabilities of the Adversaries

Adversary	Knowledge on the System
$\mathcal{A}_1$	only public knowledge
$\mathcal{A}_2$	stochastic dependency of passport number and date of expiry is known, i.e., incomplete database of BAC keys (in Germany: for each BKZ)
$\mathcal{A}_3$	complete database of BAC keys

Table 3. Eavesdropping Settings and Information for a Cryptanalytical Attack

Setting	Knowledge on the Passport Holder	Note
$\mathcal{S}_1$	issuing state	
$\mathcal{S}_2$	issuing state, photo of passport holder	
$\mathcal{S}_3$	issuing state, date of birth	
$\mathcal{S}_4$	issuing state, site of eavesdropping	relevant only for Germany
$\mathcal{S}_5$	issuing state, site of eavesdropping, and photo of passport holder	relevant only for Germany

only obtains information from the RF channel whereas setting S_2 assumes that additionally the age of the MRTD holder can be estimated by visual observation either directly or from a photo, e.g., taken in a video surveillance zone close to the inspection system. Setting S_3 acts on the strong assumption that the exact date of birth of the passport holder is known. Settings S_4 and S_5 are specific for Germany, as for this country the passport numbering scheme also depends on the issuing authority and thus generally the town of residence of the passport owner. Based on the site of eavesdropping, assumptions can be made on the issuing authority.

Table 4 gives six concrete attack scenarios, each combining an adversary from Table 2 with an eavesdropping setting from Table 3. Each scenario refers to a concrete time of the attack as the number of e-passports further increases. In our work, this concrete date is chosen to be June 1, 2007. Scenario 1 combines $\mathcal{A}_1$ and $\mathcal{S}_1$ leading to the highest complexity for both issuing states. The entropy of the date of birth denoted as H_{DB}^{G} and H_{DB}^{NL} in Scenario 1 was computed by using German demographic data [14] considering people from 18 to 80 years. In contrast, Scenario 6 combining the powerful adversary $\mathcal{A}_3$ with $\mathcal{S}_1$ needs the least key search efforts. Scenario 2 to Scenario 5 are of medium complexity acting on increasing assumptions on the capabilities and the information available for the attacker. We assume that the age of the passport holder can be guessed from

Table 4. Use Cases for Cryptanalysis in Germany and the Netherlands. The remaining entropy is estimated for each scenario.

Entropy for Germany	Entropy for the Netherlands
Scenario 1: $\mathcal{A}_1$ in $\mathcal{S}_1$ on June 1, 2007	
$H^G = H^G_{PN} + H^G_{DB} + H^G_{DE}$	$H^{NL} = H^{NL}_{PN} + H^{NL}_{DB} + H^{NL}_{DE}$
$H^G_{PN} = \log_2(10^4) + \log_2(10^5) \approx 29.9$	$H^{NL}_{PN} = \log_2(36 \times 10^6) \approx 25.1$
$H^G_{DB} \approx 14.2$	$H^{NL}_{DB} \approx 14.2$
$H^G_{DE} = \log_2(T^G_{June1,2007}) \approx 8.7$	$H^{NL}_{DE} = \log_2(T^{NL}_{June1,2007}) \approx 7.6$
$H^G \approx 52.8$	$H^{NL} \approx 46.9$
Scenario 2: $\mathcal{A}_1$ in $\mathcal{S}_2$ on June 1, 2007, Range of 10 years for date of birth: $N_{Year} = 10$ for Germany: $N_{BKZ} = 295$	
$H^G = H^G_{PN} + H^G_{DB} + H^G_{DE}$	$H^{NL} = H^{NL}_{PN} + H^{NL}_{DB} + H^{NL}_{DE}$
$H^G_{PN} = \log_2(N_{BKZ}) + \log_2(10^5) \approx 24.8$	$H^{NL}_{PN} = \log_2(36 \times 10^6) \approx 25.1$
$H^G_{DB} = \log_2(N_{Year} \times 365) \approx 11.8$	$H^{NL}_{DB} = \log_2(N_{Year} \times 365) \approx 11.8$
$H^G_{DE} = \log_2(T^G_{June1,2007}) \approx 8.7$	$H^{NL}_{DE} = \log_2(T^{NL}_{June1,2007}) \approx 7.6$
$H^G \approx 45.3$	$H^{NL} \approx 44.5$
Scenario 3: $\mathcal{A}_1$ in $\mathcal{S}_5$ on June 1, 2007, Range of 10 years for date of birth: $N_{Year} = 10$ for Germany: Local Area with 10 BKZ numbers: $N_{BKZ} = 10$	
$H^G = H^G_{PN} + H^G_{DB} + H^G_{DE}$	
$H^G_{PN} = \log_2(N_{BKZ}) + \log_2(10^5) \approx 19.8$	
$H^G_{DB} = \log_2(N_{Year} \times 365) \approx 11.8$	
$H^G_{DE} = \log_2(T^G_{June1,2007}) \approx 8.7$	
$H^G \approx 40.3$	
Scenario 4: $\mathcal{A}_2$ in $\mathcal{S}_2$ on June 1, 2007, Range of 10 years for date of birth: $N_{Year} = 10$ for Germany: $N_{BKZ} = 295$, each BKZ issues $N_P = 25$ passports per working day.	
$H^G = H^G_{PN} + H^G_{DB} + H^G_{DE}$	$H^{NL} = H^{NL}_{PN} + H^{NL}_{DB} + H^{NL}_{DE}$
$H^G_{PN} = \log_2(T^G_{June1,2007} \times N_P \times N_{BKZ}) \approx 21.5$	$H^{NL}_{PN} = \log_2(T^{NL}_{June1,2007} \times N^{NL}_{day}) \approx 20.4$
$H^G_{DB} = \log_2(N_{Year} \times 365) \approx 11.8$	$H^{NL}_{DB} = \log_2(N_{Year} \times 365) \approx 11.8$
$H^G_{DE} = \delta$	$H^{NL}_{DE} = \delta$
$H^G \approx 33.3 + \delta$	$H^{NL} \approx 32.2 + \delta$
Scenario 5: $\mathcal{A}_2$ in $\mathcal{S}_5$ on June 1, 2007, Range of 10 years for date of birth: $N_{Year} = 10$ for Germany: Local Area with $N_{BKZ} = 10$, each BKZ issues $N_P = 60$ passports per working day.	
$H^G = H^G_{PN} + H^G_{DB} + H^G_{DE}$	
$H^G_{PN} = \log_2(T^G_{June1,2007} \times N_P \times N_{BKZ}) \approx 14.6$	
$H^G_{DB} = \log_2(N_{Year} \times 365) \approx 11.8$	
$H^G_{DE} = \delta$	
$H^G \approx 26.4 + \delta$	
Scenario 6: $\mathcal{A}_3$ in $\mathcal{S}_1$ on June 1, 2007	
$H^G = \log_2(N^G)$	$H^{NL} = \log_2(N^{NL})$
$N^G \approx T^G_{June1,2007} \times N^G_{day} \approx 3.3 \times 10^6$	$N^{NL} \approx T^{NL}_{June1,2007} \times N^{NL}_{day} \approx 1.4 \times 10^6$
$H^G \approx 21.7$	$H^{NL} \approx 20.4$

a photograph with an accuracy of 10 years. Note that Scenario 2 to Scenario 5 typically have a probabilistic average success rate as the search algorithms concentrate on the most probable part of the entire key space. Therefore iterative runs with adapted assumptions might be necessary to find the BAC key. This affects especially Scenario 4 and Scenario 5 that exploit learnt stochastic properties of the passport issuing scheme. As, e.g., the number of issued passports per day may vary in practice, an uncertainty factor δ may be added here to take such deviations into account.

5 Practical Implementation on COPACOBANA

Before working out the details of the implementation we briefly introduce the underlying hardware, i.e., the cost-efficient parallel code breaker COPACOBANA[4]. The machine is built of 120 Xilinx[5] Spartan3 XC3S1000 FPGAs (Field Programmable Gate Arrays) operating independently in parallel. Instead of being soldered to one single backplane, the chips are placed on DIMMs (Dual In Line Memory Modules) in groups of six. The 20 modules are interconnected by a 64 bit data bus and a 16 bit address bus which are again connected to a controller card handling amongst others the communication with a host PC (Personal Computer) via an USB interface. A 24 MHz clock for the backplane, generated by a clock synthesizer, is used to derive a system clock by means of DCMs (Digital Clock Managers) which are part of each FPGA.

The hardware is suitable for rapidly solving parallel computation problems with low communication requirements, because the bottleneck of its architecture is the communication via the buses and to the PC. This has to be taken into account for an efficient implementation, so special care has to be taken to minimize the data traffic.

In the following we first present the general idea of how we implement the key search and then detail the content of one single FPGA and the functional units it consists of. This is followed by some statements about the execution speed and breaking the BAC with regard to some of the scenarios set up in Section 4.3.

5.1 Details of the Implementation

The key search is accomplished by segmenting the key space into practical subspaces and processing these simultaneously. Every FPGA receives the same pair of plaintext and ciphertext from the database and stores it in the corresponding registers (compare with Fig. 5), i.e., RND_{ICC} and the first 8 bytes of E_{ICC} which were previously eavesdropped, as described in Section 4.1. Dependent on the current attack scenario, e.g., from Section 4.3, the contemplable key space is divided into 120 subspaces and allocated to the same number of FPGAs, so that each unit works on a different fraction of the key space in parallel. If an FPGA is successful in finding the correct key, the respective MRZ information is output and can be stored in the database for further processing, i.e., decrypting the personal data.

A very straightforward approach of distributing the MRZ information among the FPGAs would be to provide every single MRZ to be processed by the host PC. This would involve a significant amount of data to be transferred between the PC and the COPACOBANA and thus have a severe impact on the execution time of the key search. Instead, each FPGA possesses an MRZ generator producing a new MRZ out of an assigned key space prior to each encryption. The architecture of this MRZ generator is very important for the searching efficiency

[4] See http://www.copacobana.org for more details

[5] http://www.xilinx.com

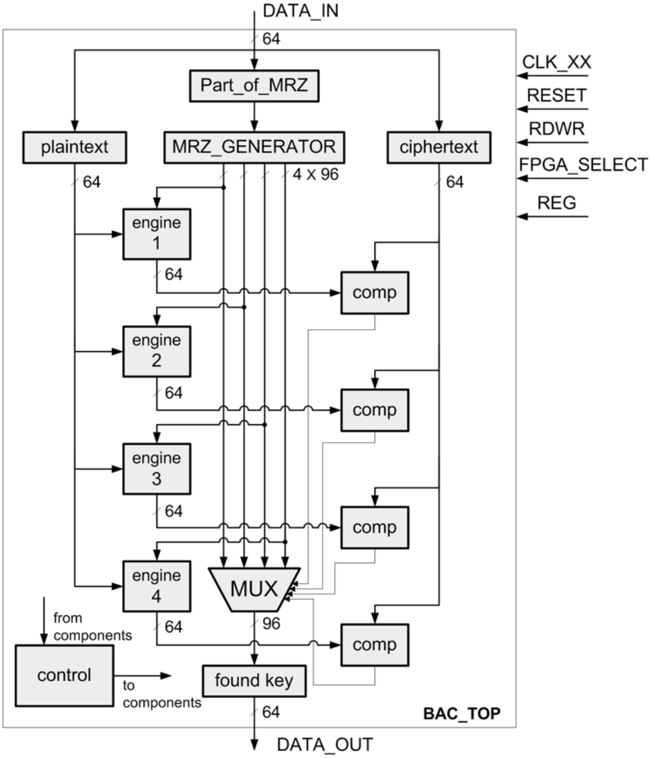

Fig. 5. Layout of a Single FPGA

of each scenario, particularly the decision which part of the MRZ information, as described in Section 2, will be fixed for each FPGA and thus stored in its Part_of_MRZ register. Therefore, the MRZ generator and hence the searching strategy can be flexibly updated for each scenario which is possible without any effort from the host PC via the USB port. Some implementation examples for partitioning the key space according to the associated scenario can be found in Section 5.2.

The main components implemented in each FPGA are four encryption engines, whose outputs are fed into four comparators for detecting a match with the default ciphertext (compare with Fig. 5). If a comparator detects that one of the four ciphertexts is identical to the one in the ciphertext register the respective MRZ information is considered as the correct key and written to the data bus.

One encryption engine, the structure of which is depicted in Fig. 6, consists of an access-key generator and a Triple-DES processor. The access-key generator is used to derive the keys for the BAC from the MRZ information, as detailed in Section 2, and thus basically performs two SHA-1 algorithms with the appropriate constants. For reducing the data traffic on the buses of the COPACOBANA, the originally 192 bits MRZ information are compressed to only 96

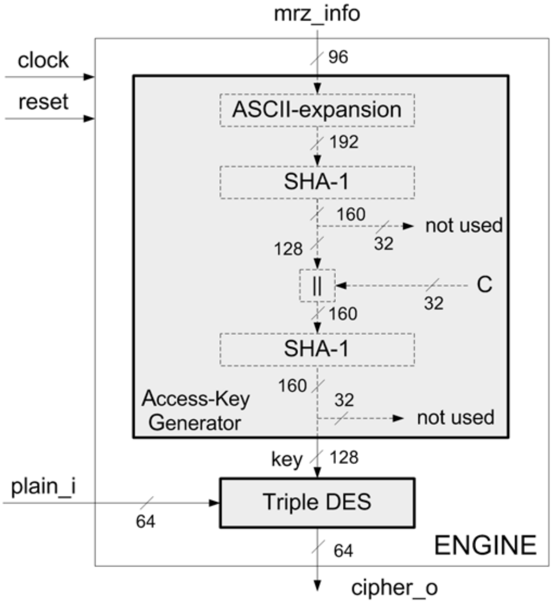

Fig. 6. Internal Structure of an Encryption Engine

bits before being sent to the FPGAs. It is the task of the ASCII-expansion unit to reconstruct the genuine MRZ information from the compressed data before the execution of the first SHA-1.

For a further speed-up, the calculation of the SHA-1, needing 80 clock cycles for one execution and therefore being the slowest part of the whole implementation, is pipelined. When the first SHA-1 has processed its data, it hands over the output value to the second SHA-1 and starts hashing the next MRZ information obtained from the MRZ generator, thus enabling simultaneous operation. Pipelining does not make sense for the Triple-DES, as its implementation, delivering a result after only 48 clock cycles, is faster compared to the SHA-1.

5.2 Practical Results

To emphasize the practical relevance of our attack, we have implemented some of the scenarios proposed in Section 4.3 in the hardware description language VHDL. The code was simulated with Xilinx Modelsim and then programmed into the COPACOBANA. All implementations have been thoroughly tested and were able to find the correct BAC key. The communication data for the tests was obtained from reading out several e-passports using the RFID reader in our laboratory.

Our implementation runs with an FPGA clock rate of 40 MHz. As the access-key generator needs 80 clock cycles to convert a MRZ into a Triple-DES key, the time needed for testing one key is $80 \cdot 25\,ns = 2.0\,\mu s$. It follows that a single FPGA consisting of four encryption engines working in parallel can check four

Table 5. Results for the Practical Implementation of some Scenarios

Issuing State:	Germany	The Netherlands
Scenario 2		
Total amount of MRZ candidates	$4.33 \cdot 10^{13}$	$2.49 \cdot 10^{13}$
Average time to find the MRZ	$\approx 9.02 \cdot 10^4\, s \approx 25\, h$	$\approx 5.18 \cdot 10^4\, s \approx 14\, h$
Scenario 3		
Total amount of MRZ candidates	$1.35 \cdot 10^{12}$	
Average time to find the MRZ	$\approx 2.82 \cdot 10^3\, s \approx 47\, min$	
Scenario 4		
Total amount of MRZ candidates	$1.06 \cdot 10^{10}$	$4.9 \cdot 10^9$
Average time to find the MRZ	$\approx 22\, s$	$\approx 10.3\, s$
Scenario 5		
Total amount of MRZ candidates	$8.85 \cdot 10^7$	
Average time to find the MRZ	$\approx 185\, ms$	

keys in $2.0\,\mu s$, i.e., two million keys per second. For all 120 FPGAs this results in $4 \cdot 120 = 480$ keys being tested every $2.0\,\mu s$, i.e., 240 million or $2^{27.84}$ keys per second.

The variable part of the implementations is the MRZ generator which hence has to be adapted to the different scenarios. As the bottleneck of the hardware is the communication via the data bus, it is advantageous to keep every FPGA occupied with key searching as long as possible. This will minimize the communication overhead and hence maximize the throughput of the machine. We found the best solution for this problem by opting for the date of birth of the passport holder as the fixed portion in an MRZ generator. This is an especially convenient situation for Scenario 2 to Scenario 5 with regard to the partitioning, because there are exactly 120 months in 10 years to be distributed to the 120 FPGAs. The expected results are summarized in Table 5.

Note that the second approach for the key search according to Section 4.2 requires only a small overhead of computational costs, i.e., four additional single DES computations, if compared to the first approach in Section 4.1 that has been the basis for our current implementation. Therefore, a realization of the second approach is also feasible with only slight modifications of the design at hand, yielding presumably the same throughput.

6 Further Directions

6.1 Software Implementation

Software implementation for cryptanalysis is an alternative choice. Fast implementations on the Pentium family require 837 cycles per SHA-1 operation and 928 cycles per Triple-DES operation [12]. Implementing a key search based on

MRZ data needs two SHA-1 and one Triple-DES, i.e., 2602 cycles in total. If pre-computed BAC keys can be used, only one Triple-DES is needed instead. Considering a Pentium clocked at 3.0 GHz, one can check about 1.15 million, i.e., $2^{20.1}$ keys per second without pre-computation and 3.23 million, i.e., $2^{21.6}$ keys with pre-computation. For the low-end scenarios involving powerful adversary $\mathcal{A}_3$, software solutions are already appropriate and probably the method of choice for implementing tracking systems. However, testing 2^{35} key candidates requires 8.5 hours without pre-computing and 3 hours with pre-computation on a single Pentium. Clusters of standard computers can further speed-up the throughput.

6.2 New FPGA Key Search Machines

The main performance bottleneck of our implementation on the COPACOBANA is the SHA-1 computation that requires 80 clock cycles per key candidate. Further, the SHA-1 determines the maximum clock frequency as it is the critical path of the overall implementation. However, as COPACOBANA was originally designed for a complete DES key search, sufficient memory for pre-computation is not available on this machine. For future designs of parallel FPGA cryptanalysis machines it is of interest whether fast on-board RAM memory can be integrated to enable key search in non-contiguous subkey spaces for determining possible speed-ups of traceability systems in hardware. Time-memory tradeoff attacks may also benefit from such a machine.

7 Conclusion

In this paper, we present the first reprogrammable hardware implementation for cracking Basic Access Control keys of the e-passport issuing schemes in Germany and the Netherlands. Our implementation is designed for the COPACOBANA that turned out to be a flexible platform for implementing probabilistic key search scenarios. The achieved throughput is 240 million, i.e., $\approx 2^{28}$ BAC keys per second. Testing 2^{35} key candidates requires 2 minutes and 23 seconds on COPACOBANA. This yields a factor of 214 if compared to a fast software implementation without pre-computation and of 74 if compared to a fast software implementation with pre-computation. These results demonstrate that key search machines are a real threat for the privacy and security of e-passport holders.

Acknowledgements. We would like to thank Tim Güneysu and Martin Novotný for their helpful and detailed explanation of how to use the key search machine COPACOBANA.

References

1. 3-millionster deutscher ePass ausgeliefert, `http://www.bundesdruckerei.de/de/presse/pressemeldungen/pm_2007_04_02.html`
2. Advanced Security Mechanisms for Machine Readable Travel Documents – Extended AccessControl, `http://www.bsi.bund.de/fachthem/epass/EACTR03110_v101.pdf`

3. Behördenkennzahl, http://www.pruefziffernberechnung.de/Begleitdokumente/BKZ.shtml
4. Behördenkennzahlen für deutsche Personalausweise und Reisepässe, http://www.pruefziffernberechnung.de/Begleitdokumente/BKZ.pdf
5. Benefits of MRTD, http://mrtd.icao.int/content/view/28/203/
6. Bundestag verabschiedet Novelle des Passgesetzes, http://www.heise.de/newsticker/meldung/90202
7. FIPS 180-1 Secure Hash Standard, http://www.itl.nist.gov/fipspubs/fip180-1.htm
8. FIPS 46-3 Data Encryption Standard (DES), http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
9. Paßgesetz PaßG, http://www.gesetze-im-internet.de/bundesrecht/pa_g_1986/gesamt.pdf
10. Privacy issues with new digital passport, http://www.riscure.com/news/passport.html
11. Avoine, G., Kalach, K., Quisquater, J.-J.: Belgian Biometric Passport does not get a pass. Your personal data are in danger!, http://www.dice.ucl.ac.be/crypto/passport/index.html
12. Bosselaers, A.: Fast Implementations on the Pentium, http://homes.esat.kuleuven.be/~bosselae/fast.html
13. Carluccio, D., Lemke-Rust, K., Paar, C., Sadeghi, A.-R.: E-Passport: The Global Traceability or How to Feel Like an UPS Package. In: WISA 2006. LNCS, vol. 4298, pp. 391–404. Springer, Heidelberg (2006)
14. Statistisches Bundesamt Deutschland. GENESIS-Online - Das statistische Informationssystem, https://www-genesis.destatis.de/genesis/online/logon
15. Finkenzeller, K.: RFID-Handbuch. Hanser Fachbuchverlag, 3rd edn. (October 2002)
16. Hancke, G.P.: Practical Attacks on Proximity Identification Systems (Short Paper). In: IEEE Symposium on Security and Privacy 2006 (2006), http://www.cl.cam.ac.uk/~gh275/SPPractical.pdf
17. Hoepman, J.-H., Hubbers, E., Jacobs, B., Oostdijk, M., Schreur, R.W.: Crossing Borders: Security and Privacy Issues of the European e-Passport. In: Yoshiura, H., Sakurai, K., Rannenberg, K., Murayama, Y., Kawamura, S. (eds.) IWSEC 2006. LNCS, vol. 4266, pp. 152–167. Springer, Heidelberg (2006)
18. ISO/IEC 14443. Identification cards - Contactless integrated circuit(s) cards - Proximity cards - Part 1-4 (2001), www.iso.ch
19. Vaudenay, S., Monnerat, J., Vuagnoux, M.: About Machine-Readable Travel Documents. In: Proceedings of the International Conference on RFID Security 2007, pp. 15–28 (2007)
20. Juels, A., Molnar, D., Wagner, D.: Security and Privacy Issues in E-passports. Cryptology ePrint Archive, Report 2005/095 (2005), http://eprint.iacr.org/2005/095.pdf
21. Kc, G.S., Karger, P.A.: Security and Privacy Issues in Machine Readable Travel Documents (MRTDs). RC 23575, IBM T. J. Watson Research Labs (April 2005)
22. Kumar, S., Paar, C., Pelzl, J., Pfeiffer, G., Rupp, A., Schimmler, M.: How to Break DES for € 8,980. In: SHARCS‘06 – Special-purpose Hardware for Attacking Cryptographic Systems, pp. 17–35 (2006), http://www.hyperelliptic.org/tanja/SHARCS/talks06/copa_sharcs.pdf

23. Kumar, S., Paar, C., Pelzl, J., Pfeiffer, G., Schimmler, M.: Breaking Ciphers with COPACOBANA - A Cost-Optimized Parallel Code Breaker. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 101–118. Springer, Heidelberg (2006)
24. ICAO TAG MRTD/NTWG. Biometrics Deployment of Machine Readable Travel Documents, Technical Report (2004), http://www.icao.int/mrtd
25. International Civil Aviation Organization. Annex I, Use of Contactless Integrated Circuit. Machine Readable Travel Documents (2004), http://www.icao.int/mrtd
26. International Civil Aviation Organization. Machine Readable Travel Documents, PKI for Machine Readable Travel Documents offering ICC Read-Only Access (2004), http://www.icao.int/mrtd
27. International Civil Aviation Organization. Machine Readable Travel Documents, Technical Report, Development of a Logical Data Structure - LDS For Optional Capacity Expansion Technologies (2004), http://www.icao.int/mrtd
28. International Civil Aviation Organization. Machine Readable Travel Documents, Supplement to Doc9303-part1-sixth edition (2005), http://www.icao.int/mrtd
29. International Civil Aviation Organization. Machine Readable Travel Documents, Doc 9303, Part 1 Machine Readable Passports, Fifth Edition (2003)
30. Robroch, H.: ePassport Privacy Attack, Presentation at Cards Asia Singapore (April 26, 2006), http://www.riscure.com