UNIVERSITY OF CALGARY

Explorations in Information Security and Privacy

CPSC 329

Assignment 2

Kenneth Sharman

ID Number: 00300185

Tutorial: T04

Instructor: Joshua Horacsek

TA: Shang Li

March 3, 2019

Question 1 - Passfaces

Passface system with a database of 100 faces. A user is randomly assigned 5 faces, to serve as their password set. Verification process consists of 5 challenges, in which he/ she must correctly identify the *one and only* correct face in a 3×3 matrix. The user must pass all 5 of these challenges in order to be authenticated.

1.1 - Entropy

To calculate the entropy of these passwords, we must first determine the number of possible password sets. Note that the order of face entry does not matter since we are dealing with a set. Further, if the order did matter, then the verification process would have to present the faces in order, as only one correct face is shown per matrix. Thus, we are interested in all possible combinations of length 5, from the set of 100 passfaces.

$$\text{Number of Password Sets} = \mathrm{C}_{(100,5)} = \frac{100!}{(100-5)!5!} = \frac{100!}{95!5!} = 75287520$$

From this we can calculate the entropy (assuming all passwords are equally likely),

$$\text{Entropy} = \left\lceil \log_2 75287520 \right\rceil = 27 \text{ bits}$$

In the context of the last assignment, we saw that 27 bits of entropy was neither super low or super high. It should be noted that the entropy classification of password strength makes the assumption that all passwords are equally likely. If the system is doing a good job at randomly assigning the passfaces, so that all assigned passwords are indeed equally likely, then it can be said that 27 bits of entropy characterizes a reasonable strength password (at least in the context of the entropy analysis we have done so far).

1.2 - Probability

I will assume that the adversary is guessing the password by making a guess for each of the five challenges. This seems to be the most logical way of guessing in the context of the question. Note that we could however (but won't), calculate the probability of the adversary simply choosing the correct five passfaces, outside of the context of these challenges.

The probability of an adversary guessing the password of a selected user can be determined by noting that each challenge is independent of the previous (or next) challenge, as in they are mutually exclusive events. Thus, the probability of correctly guessing all five challenges is,

$$P(5 \text{ of } 5) = \left(\frac{1}{9}\right) \times \left(\frac{1}{9}\right) \times \left(\frac{1}{9}\right) \times \left(\frac{1}{9}\right) \times \left(\frac{1}{9}\right) = \left(\frac{1}{9}\right)^5 = 1.694 \times 10^{-5} = 1.694 \times 10^{-3}\%$$

Thus, there is roughly a $2/100,000$ chance of guessing a users password, by guessing in each challenge. In the context of winning a small lottery, this seems like quite low odds, however in terms of computer security it may be considered relatively unsecured. It is easy to imagine that an iterative algorithm could determine the password in a very short time.

1.3 - Database public or private

The probability of correctly guessing the password would remain the same if the database was not public. This is because the number of possible 5 face sets, as well as the odds of guessing correctly, would not be impacted by making the database private. The entropy would also not change, it's just that the attacker would not be able to calculate the entropy. Thus, our quantitative classifications indicate that security would not increase if the database was not public. That being said; entropy and probability are not the only factors we can consider.

Since the password system randomly assigns 5 faces to a user, there is no way for the attacker to assess which faces would be more likely to have been picked if the selection was allowed to be done by the user (in which human tendencies could be considered). The attacker is also not able to research a particular user, and determine which type of faces he or she is likely to select, based on individual personality traits. Therefore, the attacker is really not gaining anything by having access to the database.

Finally, even if the database was private; computer security typically makes the assumption that an adversary will (one way or another) be in possession of all possible information which will help them come closer to their goal. In other words; the attacker will gain access to the database even if it is private. Thus, making the database private does not increase security. If this assumption was not made, we would have to consider how secure the database is, which constitutes an entirely different analysis and/or is a large extension to the current analysis.

1.4 - Allow 1 Failed Challenge

I am assuming that the question is interested in the probability of gaining access to the system, rather than actually obtaining the five passfaces (question asks specifically for the probability of guessing the password- not gaining access to the system by “almost” getting the whole password).

If the system allows for one incorrectly answered challenge, then the probability of gaining access will increase. This is because we *add* the probability of correctly answering exactly four out of the five, to the probability of correctly answering exactly all five challenges.

To calculate the probability of passing exactly four of the five challenges, we recognize this as a binomial experiment; there are 5 identical trials, each trial has one of two possible outcomes, the probability of success (passing the challenge) remains constant from trial to trial, and the trials are pairwise mutually exclusive. Thus, the probability of guessing exactly four out of five challenges is,

$$P(4 \text{ of } 5) = \mathrm{C}_{(5,4)}\left(\frac{1}{9}\right)^4\left(\frac{8}{9}\right)^1 = 5\left(\frac{1}{9}\right)^4\left(\frac{8}{9}\right)^1 = 6.774 \times 10^{-4} = 6.774 \times 10^{-2}\%$$

Using this result, with the probability calculated in question 1.2, we can calculate the probability of guessing the password when one fail is allowed,

$$P(4 \text{ or } 5 \text{ of } 5) = P(5 \text{ of } 5) + P(4 \text{ of } 5) = 1.694\times10^{-5} + 6.774\times10^{-4} = 6.943\times10^{-4} = 6.943\times10^{-2}\%$$

1.5 - Attacks that are effective in Passfaces compared to traditional passwords

Shoulder surfing presents a serious security risk when using passfaces. Observing a user typing in a password can provide very little information, especially if the user is a fast typer and/or their password contains a good mesh of letters, numbers, and special characters. Compare this to simply observing someone clicking on 5 images. Clearly the pictures have to be large enough that the user can make out their details, so it is easy to imagine one could obtain the password by casually taking a peek from behind, as the user selects their set of Passfaces.

Another attack that is more effective in Passfaces compared to traditional password systems is the use of screen capture. This is similar to shoulder surfing, in that the login process is being observed. With screen capture, the attacker does not need to be physically present to observe the faces selected. Comparing to a typical character-based password, which may not display the keyboard selections on the screen (although lots of systems briefly show the most recent character followed by "starred-out" characters), the selection process would be captured and the attacker would know exactly which images were selected based on the position of the mouse before the monitor display transitioned to the next challenge.

Question 2 - Picture-based Password System

Alice wants evaluate the security of a picture-based password system using 2 methods. The system has a database of 100 different pictures. The user selects 20 pictures to form a set, which is their password. There are 20 challenges, in which 2 pictures are displayed. If the user correctly answers all 20 challenges, the user is authenticated.

2.1 - Probability Calculations

Method 1: Alice will find the number of possible passwords, and use that to calculate the probability that an adversary could guess a user's password.
The number of passwords is the number of ways to choose 20 from 100:

$$\text{Number of passwords} = C(100, 20) = \frac{100!}{(100-20)!20!} = 5.360 \times 10^{20}$$

Assuming that all passwords are equally as likely (we will return to this assumption later in the question), the probability of randomly selecting the same password as the user:

$$P(\text{Random Selection}) = \frac{1}{5.360 \times 10^{20}} = 1.866 \times 10^{-21} = 1.866 \times 10^{-19}\ \%$$

Method 2: Alice will calculate the probability of impersonating the user by correctly responding to the set of 20 challenges presented by the system.
If we assume that, in the absence of any other information, each challenge has a probability of success (guessing the proper picture) equal to 0.5, then the probability of guessing the correct picture in each of all the 20 challenges is given by,

$$P(\text{20 correct guesses}) = \left(\frac{1}{2}\right) \times \left(\frac{1}{2}\right) \times \cdots \times \left(\frac{1}{2}\right) = \left(\frac{1}{2}\right)^{20} = 9.537 \times 10^{-7} = 9.537 \times 10^{-5}\ \%$$

Based on this quantitative analysis, the attacker would have a much better chance at correctly responding to the 20 challenges, as compared to randomly guessing the set of 20 pictures. This is intuitive as the challenge route means that twenty 50 : 50 picks are required, whereas correctly selecting 20 elements from a set of 100 requires the same number of selections, with a much smaller probability of success per selection.

2.2 - Compare the Usability and Security of this system with a Passfaces system described in Question 1

Usability: The Passfaces system login requires that the user only has to click five times, as opposed to twenty in the picture-based system. On the other hand, the picture-based system allows the user to add a personal touch to their password, and select the pictures themselves. This could help the user remember their set better, as a random assignment of pictures may include ones that are difficult to remember for one reason or another. Only having to choose between two pictures may be ideal for some people, as a 3×3 matrix could be somewhat daunting. Usability is a very personal topic and I am sure there are many people out there that would prefer the Passface system, and also a large number that would prefer the picture-based system. Personally, I would prefer the Passface system, as I believe it would result in a shorter login time for myself, and I don't believe I would have a problem remembering five, system assigned, pictures (although I guess I won't know until I try it!).

Security: It would be wise to calculate the entropy of the picture-based system, as we have already calculated it for the Passfaces system

$$\text{Entropy(Picture-Based System)} = \lceil \log_2 5.360 \times 10^{20} \rceil = 69 \text{ bits}$$

The entropy for the picture-based system is larger than that of the Passfaces system (69 bit compared to 27 bits) and the probability of correctly guessing the login challenges is smaller for the picture-based system than the Passfaces system (9.537×10^{-5} % compared to 1.694×10^{-3}%). It could be easy to end the discussion here, and conclude that the picture-based system is more secure than the Passfaces system, however the entropy and probability calculations are not the whole story as there are other factors that should be considered.

We saw in the previous assignment that there are many different ways to calculate entropy. The same can be said about probability. In the picture based system, the user is allowed to select twenty pictures themselves. This opens the door to many other considerations when calculating the probability, namely the introduction of human bias. People in general have certain decision making tendencies, and selecting pictures from the database would be no exception. If an attacker were to assign some sort of rank or preference order to each picture in the set, then it is clear that our probability calculation becomes not so simple. Furthermore, if the attacker has some knowledge of the user he/she is attacking then this could be used to further increase the probability of guessing the twenty pictures that the user might select.

In theory, we could increase the security, while removing the human bias, by having a system assign 20 pictures to a user and implement the 20 challenge authentication process. The problem with this is that it would drastically decrease usability. The user would be forced to not only remember 20 "random" pictures (and pictures in general may be less memorable than faces) but also be subjected to the longer login processes. Clearly there is a delicate balance between security and usability- both of which are subject to a certain amount of subjectivity, making it difficult for a system manager to decide on the ideal or optimal password system.

2.3 - Algorithm to learn a user's passwords if unlimited attempts were allowed

Assume that the system ends the authentication process as soon as an incorrect selection is made. With this assumption, an attacker can very quickly determine the password set. Consider the first challenge. If the attacker guesses correctly, then they will know one of the pictures in the set. If they guess incorrectly, then they also get this information, since there are only two choices- it's the other picture. If the attacker continues to try this challenge based authentication process, he/she will eventually see all the pictures in the password set. The question then becomes, how many challenges are required on average to see all 20 pictures in the set.

For simplicity, let's assume that the attacker always fails the first challenge. This is the worst case analysis.

First picture: Attacker gets it on the first attempt (as mentioned above, this is independent of his/her challenge selection).

Second picture: There is a $p = 19/20$ chance of seeing a new (unknown) picture from the password set. If we label a 'success' as seeing a new picture, then the average number of trials, X, for a success to occur is given by the expectation value of the *Geometric Distribution.*

$$E(X) = \frac{1}{p} = \frac{20}{19} = 1.05$$

Third picture: This is the same as the previous case, except the probability of seeing a new picture is $p = 18/20$. The expected number of additional trials to get the third picture is again given by the expectation value of the Geometric Distribution: $E(X) = \frac{1}{p} = \frac{20}{18} = 1.11$

We can continue this calculation the same way, until we are looking for the final picture. The probability is then $p = 1/20$ and $E(X) = 20$.

Summing the expected number of trials (challenges) together gives:

$$\sum_{i=1}^{20} \frac{20}{i} = 20 \sum_{i=1}^{20} \left[\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{20}\right] = 71.95 =\approx 72$$

A worst case estimate for the number of guesses required by the attacker is 72 attempts (worst case because we assumed the attacker would always fail the first challenge).

Now, in the beginning of this analysis we stated that the authentication system would end as soon as an incorrect selection was made. This is quite insecure, as that information tells the attacker what the correct choice was. If we assume that the challenge based authentication process continues for all twenty challenges, regardless of what is being answered correctly, then things get a little more complicated. I am unsure how to proceed with the analysis, but it is clear that a frequency based approach would be required. Since a correct picture (one that is in the password set) is always presented, over a large number of attempts it should become clear which pictures are in the password set, simply based on the fact that they show up more often than pictures that are not in the set. If an attacker were to keep track of each frequency of occurrence for a given picture, then he/she would become increasingly confident on the contents of the password set as the number of trials is increased.

Question 3 - RFID with Exclusive Or

Consider the following protocol for an RFID to authenticate and RFID tag. The reader generates a random 32-bit challenge 'x' and transmits $y = x \oplus s$ to the tag. The tag computes $z = y \oplus s$ and sends it to the reader. The read authenticates the tag is $z = x$.

3.1 - Eavesdropper can recover the key by observing a single transaction

Let:

$$y = 0x3344ffac \qquad \text{and} \qquad z = 0x1100dd0d$$

The inverse function to XOR is also XOR so we can compute the secret key, $s = y \oplus z$, as follows (conversion to binary is not explicitly shown as this was covered in prerequisite courses):

$$\begin{array}{r@{\,}l} y = & 0011\ 0011\ 0100\ 0100\ 1111\ 1111\ 1010\ 1100 \\ z = & 0001\ 0001\ 0000\ 0000\ 1101\ 1101\ 0000\ 1101 \\ \hline s = y \oplus z = & 0010\ 0010\ 0100\ 0100\ 0010\ 0010\ 1010\ 0001 \end{array}$$

Now, for the given z, we know that the challenge is $x = 0x1100dd0d$. We can show that s is the correct key by showing how it completes the "communication loop".
The reader transmits:

$$\begin{array}{r@{\,}l} x = & 0001\ 0001\ 0000\ 0000\ 1101\ 1101\ 0000\ 1101 \\ s = & 0010\ 0010\ 0100\ 0100\ 0010\ 0010\ 1010\ 0001 \\ \hline y = x \oplus s = & 0011\ 0011\ 0100\ 0100\ 1111\ 1111\ 1010\ 1100 \end{array}$$

The tag responds with:

$$\begin{array}{r@{\,}l} y = & 0011\ 0011\ 0100\ 0100\ 1111\ 1111\ 1010\ 1100 \\ s = & 0010\ 0010\ 0100\ 0100\ 0010\ 0010\ 1010\ 0001 \\ \hline z = y \oplus s = & 0001\ 0001\ 0000\ 0000\ 1101\ 1101\ 0000\ 1101 \end{array}$$

Comparing the challenge to the response, we see that

$$x = z = 0001\ 0001\ 0000\ 0000\ 1101\ 1101\ 0000\ 1101 = 0x1100dd0d$$

which indicates that we have found the secret key. Note that this is not a proof that the inverse function of XOR is also XOR, however the calculations serve as an example calculation to show that this process works.

3.2 - Variation of the authentication protocol

The reader and the tag share two different keys, s_1 and s_2. The reader sends challenge $y = x \oplus s_1$ and the tag responds with $z = x \oplus s_2$ after recovering $x = y \oplus s_1$. Can a passive eavesdropper learn the secret keys from observing a single execution of the protocol?

During a single execution of the protocol, the eavesdropper is able to intercept both the value sent by the reader and the value sent by the tag,

$$y = x \oplus s_1 \qquad \text{and} \qquad z = x \oplus s_2$$

The attacker cannot manipulate these values to obtain s_1 and s_2 because x is also unknown. The simplest way to come to this conclusion is to realize that despite the many XOR identities, we have a system of 2 equations with 3 unknowns. Without multiple executions, there is no way to solve this system.

3.3 - Multiple Executions

If the attacker is able to observe multiple executions of the protocol, then s_1 and s_2 will eventually be determined. We will use the properties of the exclusive or function [1] to show how this can be done. Note we assume that the length of all variables is the same.

Property 1:

$$\text{If } A \oplus B = C \qquad \text{then} \qquad C \oplus B = A \qquad \text{and} \qquad C \oplus A = B$$

Property 2:

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C$$

Property 3:

$$A \oplus B = B \oplus A$$

Property 4:

$$A \oplus A = 0$$

Property 5:

$$A \oplus 0 = A$$

Now, during a single execution of the protocol, the attacker can obtain:

$$y = x \oplus s_1 \qquad \text{and} \qquad z = x \oplus s_2$$

Using property 1 on both of these equations, we can get

$$s_1 = y \oplus x \qquad \text{and} \qquad s_2 = z \oplus x$$

Combining the equations, then using Properties 2, 3, 4, then 5

$$\begin{aligned} s_1 \oplus s_2 &= (y \oplus x) \oplus (z \oplus x) \\ &= y \oplus (x \oplus z) \oplus x \\ &= y \oplus (z \oplus x) \oplus x \\ &= (y \oplus z) \oplus (x \oplus x) \\ &= (y \oplus z) \oplus 0 \\ &= y \oplus z \end{aligned}$$

We see that after the first execution of the protocol, an attacker would be able to determine the value of $s_1 \oplus s_2$ by calculating $y \oplus z$. At this point the attacker could assemble a list of all possible s_1 and s_2 that yield the known value of $s_1 \oplus s_2$.
Now, in the second execution of the protocol, the attacker would calculate:

$$y \oplus (s_1 \oplus s_2) = z = x \oplus s_2$$

Using the list of possible s_2 values (compiled in the previous step), a list of possible x values can be calculated using $x = z \oplus s_2$. Similarly, the attacker can calculate,

$$z \oplus (s_1 \oplus s_2) = y = x \oplus s_1$$

and use the list of possible s_1 values to create a list of possible x values. Here is the key step; if there is an x that is not in both of the possible x lists, then the corresponding s_1 or s_2 can be eliminated from the possible values that result in the correct value of $s_1 \oplus s_2$.

Repeating this process (listening to multiple executions of the protocol) until the list of possible s_1 and s_2 values has reduced to single values would result in the attacker being able to successfully learn the secret keys.

Question 4 - RFID with Hashing

Consider an RFID authentication system that can be used in two modes: scanning mode (inventory) and individual mode (checkout).

Protocol 1: Reader sends a random challenge r. Tag replies with h(r|ID), ID

Protocol 2: Reader sends a random challenge r. Tag replies with h(ID|k_{ID}) $\oplus$ r, ID

Attack 1: An adversary tampers with tags' responses during a scanning round, with the goal of corrupting the shop's database.

Attack 2: An adversary tampers with the response during checkout, with the goal of paying less for the item.

4.1 - Attack 1 on Protocol 1

Create a new tag ID and pretend to be this new tag using some sort of electronic device, for example, a phone. In the scanning mode, the reader broadcasts the challenge to all tags in a short range from the reader, so our imposter tag will have no problem receiving the challenge. Protocol 1 simply responds with the hash of the challenge concatenated with the ID. If we assume that the attacker has access to the hash function, then there is no issue with hashing the challenge and fake ID.

The issue with this attack is seen when the reader receives the response from the tag. If the system is set up so that any new ID will simply be added to the inventory, then indeed the attack will accomplish the goal of corrupting the database, as invalid products can be added to the inventory. The more likely scenario is that the reader will attempt to increase the on-hand count of the product corresponding to the ID (since this is used for inventory), and since the imposter ID is not already in the inventory, the attack will fail. Thus, the success of this attack depends on how the reader/ inventory system is implemented.

4.2 - Attack 2 on Protocol 1

The attacker could use a phone to send a challenge to the tag of a low cost item. The tag would respond with the hash value of r|ID and the ID. The ID is easily obtained, and if we assume that the hash function is known, then the attacker has all the information required to impersonate this item.

Next the attacker would have to disable the response from the true tag (the more expensive item), or sufficiently delay it. In the checkout the reader would send a challenge, which the phone could respond to before the tag of the item to be purchased. The checkout process would display the lower cost item (which the phone was impersonating) and the attacker would successfully achieve their goal.

4.3 - Attack 1 on Protocol 2

This is a subtle difference between Attack 1 on Protocol 1 and Protocol 2. Let's put aside the case where the reader will simply increase the on-hand count of an item already in the system. If this is the case, then imposter items cannot be added to the system by responding to challenges.

If products can be added to the inventory during the scanning mode, then Protocol 2 introduces an extra layer of security, in the form of the secret key. We are assuming the same scenario as discussed in 4.1. This time however, the fake ID is concatenated with an unknown secret key and XOR with the challenge. The reader examines the response and determines if the hashed value is correct for the provided ID, using the secret key. Given that some hash functions have collisions, it is possible to use brute force (as in respond to multiple executions of the protocol) to give a correct response to the challenge. Clearly the probability of a successful attack is much lower than using Protocol 1.

If the hash function has no collisions, then the secret key would have to be determined.

4.4 - Attack 2 on Protocol 2

The attacker would need to obtain the hashed value of ID$|k_{ID}$, and the ID. Again, a phone could be used to send various challenges to a low cost item. The tag would respond with [h(ID$|k_{ID}$) $\oplus$ r, ID]. Clearly the ID could be obtained, since it is simply the second element in the response. If we assume that the hashed value and challenge have the same length, then the hashed value could be obtained by performing XOR with the challenge (which is known since the phone created it). Now the attacker has all the information required to achieve the goal of paying less at checkout.

If the true tag of the item to be purchased is tampered with, so that it cannot respond, then the phone could respond to the challenge sent by the checkout reader, by calculating h(ID$|k_{ID}$) $\oplus$ r, and responding with the correct value to impersonate the low cost item.

We can see that protocol 2 is slightly more involved than the first, since the XOR operation would need to be performed.

Question 5 - Canadian E-Passport

5.1 - Does Canadian E-Passport store biometric data?

According to the government of Canada [6] the E-Passport "is also known as a biometric passport". It stores the information found on page 2 of the Canadian passport which includes the persons surname, given name, date of birth, sex, and a digital picture of the persons face. Thus, the only biometric the RFID chip stores is the digital picture of the passport holders face. We recognize this as a biometric since the picture is used for authentication purposes and is most certainly based on a physical characteristic.

0.1 5.2 - Outline two important security and privacy issues related to biometrics

An important privacy question many people would likely have regarding ePassports would be the ability to track an individual. Fortunately, Canadian ePassports are passive (meaning it doesn't have any power source) and they cannot send information over long distances, even when they are powered [6]. Thus, this is not a concern for Canadian citizens.

A major concern can be seen with the data leakage threats. E-Passports are vulnerable to skimming [2], which is when an unauthorized entity secretly reads the contents of the passports. There is sensitive information stored on the RFID chip, and it would be a violation of privacy if an unauthorized entity were to read and collect this data (could lead to Identity theft). The countermeasure implemented by the Canadian ePassport is that the passport must be within 10cm of a reader, and the birthday of the passport holder, the passport expiration date and number must be provided to the reader. This countermeasure is dubbed BAC or Basic Access Control [5].

An additional skimming threat is that an eavesdropper could listen to the communication between the passport and an authorized reader. BAC also protects against this sort of wireless communication intercept by encrypting all data transferred between the passport chip and the reader. The encrypted data cannot be "easily" decrypted by a radio frequency eavesdropper [5].

0.2 5.3 - RFID Chip

A key difference between different types of RFID tags is the frequency they are tuned to. The tag and the reader must be tuned to the same frequency in order to communicate with each other. The most common frequencies are [4]

- Low Frequency (Lf), 125-134 kHz
- High Frequency (HF), 13.56 MHz
- Ultra-High Frequency (UHF), 433 MHz and 860-960 MHz

In general, the larger the frequency, the longer communication range that is offered by the tag and reader combination. An example of RFID tags in the retail sector can be seen with CDs, in say HMV. These chips operate at 13.56 MHZ, which is the same frequency used by the chips in the Canadian E-Passports. One reason they are used with CDs is because High Frequency tags work well on objects made of metal. While HF RFID tags have a maximum range of about 1 meter, the chip on the Canadian Passport is specifically designed to have an operating range of 10 cm. This is a key difference between such a retail RFID tag and the one used in the E-Passport.

In other applications it might make more sense to employ a UHF RFID. These have a high data transfer rate and are suited for many items at once. An application ideally suited for this kind of chip would be in the shipping industry. The inventory of a shipment could be taken very quickly by scanning a skid of boxes, rather than opening the shipment and individually scanning all the bar codes. The long range offered by UHF RFIDs is not well suited for passports, since it could potentially transmit sensitive information right "into the hands" of an attacker.

All RFID tags use the same principle of communicating between the tag and the reader using electromagnetic waves, however there are most certainly vasts differences between the various varieties. One of the keys to the success of the Canadian E-passport is that it uses passive authentication. This is required to show that the data on the passport hasn't been modified. Without it, the whole concept of the E-Passport would be useless, as it would do no better to authenticate a persons identity than the traditional (old-school) passport. For most retail situations (such as a clothing store), this level of authentication would be overkill and we would not expect to see it used.

0.3 5.4 - Does the cryptography conform to the ICAO standard?

Canadian E-Passports comply with ICAO Doc 9303 specifications for Machine Readable Travel Documents [3].

0.4 5.5 - Report one security or privacy related incident related to electronic passports

Just a few months ago, on November 30, 2018, Marriott International announced that roughly 5 million *unencrypted* passport numbers, names, and addresses were accessed by hackers. The data information belongs to Marriott Hotel customers. The attack began in 2014 when information in the Starwood guest reservation system was accessed by hackers. A few years after this, in 2016, Marriott acquired the Starwood reservation system. The breach quietly continued, as it was overlooked during this acquisition period, until 2018 [7]. The articles referenced did not state how the attack was detected.

The attackers are not known, however the style and timing of the attack indicate to some experts that the attack was directed by the Chinese Ministry of State security. Regardless of whether or not these suspicions are true, it is easy to speculate that the motivation for the attack involved building profiles and tracking people of interest, as the accessed information would provide knowledge as to where and when particular individuals would be traveling and staying at these hotels.

Marriott did not disclose how the attack was identified, however they have stated that they would provide reimbursement for passport replacement fees, if they conclude that the breach has lead to an occurrence of fraud for a particular guest.

A consequence of this attack is that it has initiated a larger discussion regarding the level of accountability required for companies who have sensitive information, such as Marriott did in this case. As we implement more and more technology into our everyday lives, data storage and protection becomes increasingly important and it is essential that those in possession of sensitive information must be held to a certain standard. It is unfortunate that large data breaches, such as this, seem to be required to highlight the importance of data security, but that is the general trend with Infosec- it is usually an afterthought.

Links:

https://www.washingtonpost.com/technology/2019/01/04/marriott-hackers-accessed-more-than-million-passport-numbers-during-novembers-massive-data-breach/

https://www.forbes.com/sites/davidvolodzko/2018/12/04/marriott-breach-exposes-far-more-than-just-data/

Evidence that Source are Credible:

A piece of evidence that the article sourced from the Washington Post has a certain amount of credibility is that the author, Peter Holley, can easily be verified to be a technology reporter for the Washington Post. An online search reveals that he has done work for many reputable publishers, such as the Los Angeles Times, Time Magazine, and the Business Insider to name a few. The fact that his works can be seen in these top name publications establishes a certain amount of credibility.

The evidence supporting the credibility of the second article is that the publisher, Forbes, is known around the world in the business and finance sectors. Navigating around the website indicates that the site is not an imposter site, as all the linked articles appear valid and written by genuine authors. It would not be in the best interest of Forbes to publish a story that fabricated such serious allegations towards a huge company such a Marriott. All these factors indicate that the articles content is indeed credible.

References

[1] CS 3343. Analysis of algorithms. http://www.cs.utsa.edu/ wagner/CS3343/xor/xor.html. Accessed: Feb 16, 2019.

[2] David Molnar Ari Juels and David Wagner. Security and Privacy Issues in E-passports. https://eprint.iacr.org/2005/095.pdf. Accessed: March 3, 2019.

[3] ICAO. Doc 9303 machine readable travel documents. www.icao.int. Accessed: Feb 17, 2019.

[4] RFID Insider. Which rfid frequency is right for your application? blog.atlasrfidstore.com. Accessed: Feb 17, 2019.

[5] Covernment of Canada. Technical information about the Canadian ePassport. www.canada.ca. Accessed: March 3, 2019.

[6] Government of Canada. The epassport. www.canada.ca. Accessed: Feb 17, 2019.

[7] The Washington Post. Marriott: Hackers accessed . . . . www.washingtonpost.com. Accessed: March 3, 2019.