Assignment 3

Dan Reynolds
djreynol
20473104

1.

b)

If a one-time pad key is used more than once, it introduces a security vulnerability that allows someone who has both plaintexts encrypted using the same key to decrypt them both and read the plaintexts.

This was accomplished by using the fact that given an encryption key k, and plaintexts p_1, p_2 we know that the resultant cipher texts are:

$$c_1 = p_1 \oplus k$$
$$c_2 = p_2 \oplus k$$

Given that k is the same, we have:

$$\begin{aligned} R &= c_1 \oplus c_2 \\ &= p_1 \oplus k \oplus p_2 \oplus k \\ &= p_1 \oplus p_2 \end{aligned}$$

The result R we get from XORing the two ciphertexts together is the XOR of the two plaintexts together.

Given this knowledge, I then used an approach called crib-dragging, described in this article https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma

That basically involves guessing a word likely to be in one text, and XORing it with R at different locations. For example, if guessing "␣the␣" at a location in R returns "Bohem" as it did in my text, you know that one of the plaintexts contains "␣the␣" and at a matching location, the other contains "Bohem". From "Bohem" I was able to guess "Bohemian" and expanded my results.

It is called dragging because if you did not find "Bohem" right away, you would drag the guess word "␣the␣" across the XOR'ed text, starting at different indices until you found a likely match. If no result is ever returned that seems to match with "␣the␣" then you can try other common words and slowly build up the original plaintexts.

2.

a)

I am tracking whether someone has the occupaation of Staff or not, so my tracker is:

T = `WHERE OCCUPATION = "Staff"`

We know that the tracker matches between $\frac{N}{8}$ and $\frac{7N}{8}$ as required since we know the company hires an equal number of Specialists and Staff employees.

The three queries are therefore:

1. `SELECT SUM(SALARY) FROM EMPLOYEE WHERE OCCUPATION = "STAFF" OR NAME = "Lucille"`
2. `SELECT SUM(SALARY) FROM EMPLOYEE WHERE OCCUPATION != "STAFF" OR NAME = "Lucille"`
3. `SELECT SUM(SALARY) FROM EMPLOYEE`

Given these three queries, we know that Lucille is included in both 1 and 2, meaning that queries 1 + 2 - 3 will only be different by Lucille's salary, since the only common salary in 1 + 2 was Lucille's, and otherwise 1 + 2 were disjoint.

Therefore 1 + 2 - 3 leaves only Lucille's salary as desired.

b)

Given this additional constraint, I would learn Rachel's salary using a modified binary search approach.

I would use 4 queries each iteration of my search:

```
SELECT COUNT(*) FROM EMPLOYEE WHERE SALARY > 100000
SELECT COUNT(*) FROM EMPLOYEE WHERE SALARY > 100000 AND NAME != 'Rachel'
SELECT COUNT(*) FROM EMPLOYEE WHERE SALARY <= 100000
SELECT COUNT(*) FROM EMPLOYEE WHERE SALARY <= 100000 AND NAME != 'Rachel'
```

I use 100000 as my starting point because the range of potential salaries is 0-200000 and a binary search begins at the midpoint.

If there is a difference in the COUNT returned for my two guesses of a salary greater than 100000, I know Rachel is in that half. If there is also a difference in my two queries of less than or equal to 100000 then I know Rachel is in both halves and must have a salary of 100000. If there is only a difference in one half, then I branch that way.

To avoid the problem of falling below a query result size of $\frac{N}{8}$ or over $\frac{7N}{8}$, I can just do the following 2 queries:

```
SELECT COUNT(*) FROM EMPLOYEE WHERE OCUPPATION = "Staff" AND NAME != 'Rachel'
SELECT COUNT(*) FROM EMPLOYEE WHERE OCUPPATION = "Staff"
```

I now know Rachel either has the Staff occupation or not and can either use an OR or AND condition in my SELECT statements to include or remove the $\frac{4N}{8}$ employees that are not Rachel's profession in order to keep the size of the result returned from my queries in the range $\frac{N}{8}$-$\frac{7N}{8}$. Since this group does not contain Rachael, it keeps me in the correct range without affecting the way I branch using the above 4 queries.

I can then continue my binary search until I find her exact salary in approximately $4 \log 200000$ queries.

c)

The table is not 3-anonymous, because not every quasi-identifier combination in the table has at least k - 1 = 3 - 1 = 2 records from which it cannot be distinguished. For example, the quasi-identifier of Name, Birthdate and Occupation with `Name = *, Birthday = 74*, Occupation = Specialist` has no other row with this combination meaning that the table is not 3-anonymous, as there is a quasi identifier combination without at least 2 other records sharing that identifier.

New table:

Name	Birthdate	Occupation	Allegiance
*	7*	Specialist	Quendor
*	8*	Specialist	Quendor
*	7*	Staff	Quendor
*	7*	Specialist	Antaria
*	7*	Staff	Antaria
*	7*	Specialist	Quendor
*	8*	Specialist	Antaria
*	8*	Specialist	Antaria
*	7*	Staff	Kovali
*	7*	Staff	Kovali

This table is 3-Anonymous, and can easily be checked:

Quasi identifier row count: `*, 7*, Specialist` = 3 `*, 7*, Staff` = 4 `*, 8*, Specialist` = 3

The l-diversity is determined by the minimum value such that for each unique quasi-identifier, there are at least l different values in the sensitive columns for that quasi-identifier.

Quasi identifier different sensitive value count: `*, 7*, Specialist` = 2 `*, 7*, Staff` = 3 `*, 8*, Specialist` = 2

Therefore the table has an l value of 2 and is 2-diverse since each unique quasi-identifier has at least 2 different values for the identifier's sensitive information.

3.

a) A U.S. student is now allowed to copy contents of a television show on Blu-Ray for criticism or comment as part of educational purposes rather than having to use a screen-captured version as long as it is on a lawfully-made Blu-Ray and screen capturing is believed to not be sufficient enough for the educational purpose.

b) Classes 11-15 investigated circumventions of access controls for mobile connectivity devices such as cellphones to allow them to connect to different carriers, rather than the one they bought if from, who may have subsidized the cost.

Three parties that opposed the exemptions included:

- TracFone: Opposed cellphone unlocking in class 11 because of the concern that this could promote the buying of phones at a subsidized cost and re-selling of them for profit, since the buyer is not restricted to sell the phone on the same carrier.
- GM: Opposed class 13 for the broader class of mobile connectivity devices because they wanted to exclude those devices in automobiles, so that their devices could not be used in other automobiles. The companies wanted the ability to lock these devices to specifically the company that provided the devices in the automobiles.
- Alliance of Automobile Manufacturers (AA): The AA opposed class 15, which specifies for the unlocking of all wireless "consumer machines", which covers most Internet of Things devices such as appliances and smart meters. AA argued that the category is too "broad and diverse" and that if adopted, it might apply to smart devices that apply to automobiles as well, which is their primary concern.

Like GM, they believe that unlocking should not be permitted for automobile-based wireless devices.

c) Security researchers are given exemptions on lawfully acquired devices for the purpose of good faith security research, although the exemptions do not take effect for 12 months until after the date of this regulation, with the exception of voting machines.

The systems and devices that are granted these exemptions for the purpose of detecting security flaws and vulnerabilities must be primarily designated for use by individual consumers, like cell phones versus nuclear power plants. Categories listed include motorized land vehicles, and medical devices designated for partial implantation in patients or systems for personal monitoring.

d)

While it is legal under this ruling for the professor to bypass copyright protection for a DVD and copy content in a limited educational purpose, it is not legal to distribute software for copying content on protected DVDs on the Internet, even with the intention of having it used for educational purposes because as specified in the ruling, the trafficking and manufacturing of products or services used to circumvent technological measures that protect exclusive copyrights is illegal and the Librarian cannot make exceptions for such services. Therefore while it is legal to do it in class under the right circumstances, there is no exemption making it legal to traffic and manufacture software that allows others to copy content on protected DVDs.

4.

The number of oracle calls is dependent on the the number of blocks N, number of bytes in the block b, and the number of possible values of each byte, W.

The average number of calls is $\frac{NbW}{2}$ since on average it takes $\frac{W}{2}$ calls to guess the correct value such that the intermediate value will XOR with the guessed value in the preceding block to produce the correct padding.

In the worst case, this takes all W guesses, giving NbW total calls in the worst case.

5. Adding a layer of authentication before the encryption process is one tool that can help to fix this vulnerability. The authentication layer provides both confidentiality and authenticity, as the cookies are only viewable by users with the correct authentication. With an authenticity factor applied to the plaintext before encryption, the side channel is closed because the server will not XOR the attacker's guess and affirm or reject their attempt at generating the correct padding unless the authenticity layer is valid. This way, a different message is returned if the ciphertext sent to the server is not authenticated and an attacker will learn nothing about the encryption.

The web server is now only vulnerable to users that are trying to attack the server and have proper authentication. But since the server now knows who is authenticating, it can deny that user access after a certain number of attempts, or use a similar technique to limit a valid user's attempts to determine the padding of the protocol.

6. In the new padding scheme, the same number of bytes are used to indicate padding, but only the last byte contains the number of padding bytes as its value, and the rest are 0.

Section 3.1:

Line 5.b) must change because in 5.a)-5.b) it determines the number of padding bytes in the block and on 5.b) outputs the intermediary values for each padding byte by XOR'ing each byte in r with n, the number of padding bytes.

In the new scheme this is wrong, as only r_b, the last byte of r has an associated plaintext value of n, so its intermediary value is $r_b \oplus n$. In the new scheme, for all the other padding bytes it should now just output the r value as the intermediary value, as the value of the padding for the other bytes is 0. Therefore line 5.b) becomes:

if O(r|y) = 0 then stop and output $(r_{b-n+1}...r_{b-1})) \mid (r_b \oplus n)$

Section 3.2:

The difference in this section is first on line 1, which sets the correct guesses for each r_k for which we have already found its associated intermediary value in the block a_k. While in the old scheme, we wanted to set each of their assocaited plaintext values equal to the number of padding bytes (b - j + 2), now only the last byte's plaintext is this value, and the other padding bytes must have plaintext value 0.

Therefore we must now expand line 1 to two separate lines:

take $r_b = a_b \oplus (b - j + 2)$
take $r_k = a_k$ for k = j,..,b-1

Then later on like 5, we must change what we output. In the old scheme, the intermediary value a_{j-1} for r_{j-1} would be $r_{j-1} \oplus i \oplus (b - j + 2)$ because a_{j-1} must have had as its plaintext value equal to the number of padding bytes. But now we know that it instead has the value 0, so line 5 becomes:

output $r_{j-1} \oplus i$

since a_{j-1} must have had value 0 and does not need to be XOR'ed with a plaintext padding value.