Home Assignment 2

Advanced Web Security

2017

B-assignments

For grade 3, complete the three B-assignments below and solve them in groups of ≤ 2 students.

B-1 DC-nets and mixnets are the two fundamental existing techniques for anonymous communication. Mixnets have received much more attention and are also in focus in this course. However, the treatment of anonymity would not be complete without having mentioned at least the basic ideas behind the DC-nets. Read about the Dining Cryptographers (DC) problem. It was first propoesd in

> D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. In *Journal of Cryptology*, 1(1), pp. 65–75, Springer-Verlag, 1988.

but for this course you only have to read the introduction. A perhaps more clear explanation can be found on the Wikipedia entry:

> `https://en.wikipedia.org/wiki/Dining_cryptographers_problem`

Assume that you are sitting around a table with two friends who happen to be named Alice and Bob. You decide to implement the DC-protocol referred to above, but instead of sending a single bit, you wish to allow sending 16-bit anonymous messages at the same time. This can be accomplished by just running 16 parallel instances of the protocol. Implement a program that takes as input

- Your shared 16-bit secret with Alice (SA).
- The broadcasted data sent by Alice (DA).
- Your shared 16-bit secret with Bob (SB).
- The broadcasted data sent by Bob (DB).
- A 16-bit message M that you *may* wish to send anonymously.
- A bit b, such that if $b = 0$ you do not wish to send the message and if $b = 1$ you wish to send the message.

The program output should be

- If $b = 0$, your broadcasted data in hex format, immediately followed by the anonymous message sent by some other party (0000 if no anonymous message was sent).
- If $b = 1$, your 16-bit broadcasted in hexadecimal format.

Example:

SA	SB	DA	DB	M	b	Program output
0C73	80C1	A2A9	92F5	9B57	0	8CB2BCEE
27C2	0879	35F6	1A4D	27BC	1	0807

Assessment:

- Upload your code to Urkund, jonathan.sonnerup.lu@analys.urkund.se. Only one student per group must do this.
- There will be a Moodle question following the problem statement above. Both students must finish the Moodle quiz. There will be a test quiz on Moodle where you can try your implementation as many times as you like. The test quiz will not be graded.

B-2 The CIA team led by the CTC director Carrie Mathison have received valuable intel on ongoing terrorism activity. The infamous terrorist Abu Nazir have been communicating through a Mix network[1], a system that allows for anonymous communication. The security team have been analyzing the network traffic to and from the Mix, but they are having trouble making sense of all the data. You, a recently graduated engineer with a passion for security is hired to assist the CIA in this delicate matter. Having just read about the disclosure attack on Mixes, you want to use this attack to disclose the IP addresses of Nazir's partners.

The format of the log files is a standard `.pcap` format (viewable in Wireshark and similar) containing enough information to find the partners. In general, to find m disjoint sets out of a large number of sets is an NP problem[2]. In this case, we assume that when Nazir sends data to a partner, it is to one and only one partner in a batch. In the same batch, no other sender will send data to any of Nazir's partners. Thus, we can be sure that every new set we find which is disjoint to the previously saved sets can safely be saved. The Mix waits a fixed number of minutes before sending out the complete batch. While the Mix is sending out the batch, there is no data coming into the batch.

You are given example code in Python[3]. For Java, you may use the library `pkts.io`[4].

Implement a program that takes as input:

- Abu Nazir's IP address.
- The Mix's IP address.
- The number of partners to Nazir.
- A `.pcap` file which contains the sniffed network traffic to and from the Mix.

The program output should be the integer sum of the partners' IP addresses.

Example:
Assume we have disclosed all 3 of Nazir's partners' IP addresses, which are:

```
85.14.156.21
47.56.124.15
210.25.145.218
```

Interpreting the IPs as 4-byte (big-endian) integers in hexadecimal:

```
55 0e 9c 15 -> 0x550e9c15 = 1427020821
2f 38 7c 0f -> 0x2f387c0f = 792230927
d2 19 91 da -> 0xd21991da = 3524891098
```

The sum of the IPs is 5744142846.

[1] `https://en.wikipedia.org/wiki/Mix_network`
[2] `https://en.wikipedia.org/wiki/NP_(complexity)`
[3] `https://github.com/eit-lth/Advanced-Web-Security_EITN41/tree/master/2%20Anonymity/code/`
[4] `http://www.aboutsip.com/pktsio/`

Note: The integer representation of an IP address is valid, for example, the IP address of `google.se` is `173.194.79.94`, which converts into the integer 2915192670, and you may access Google by entering `http://2915192670` in your browser.

Assessment:

- Upload your code to Urkund, jonathan.sonnerup.lu@analys.urkund.se. Only one student per group must do this.
- There will be a Moodle question following the problem statement above. Both students must finish the Moodle quiz. There will be a test quiz on Moodle where you can try your implementation as many times as you like. The test quiz will not be graded.

B-3 The Tor design paper includes a description of hidden services. Read about Hidden Services in the Tor paper (Section 5). There have been some research results on how to break the anonymity of Hidden Servers in Tor. The first paper detailing how such an attack would proceed, together with proposed countermeasures is

> L. Overlier and P. Syverson - Locating Hidden Servers. SP '06 Proceedings of the 2006 IEEE Symposium on Security and Privacy, 2006.

Read this paper and understand how the attacks work.
Assessment:

- There will be 4 Moodle questions related to this assignment. Each question will give you 0 - 0.5 points. If you work in a group, both students must answer the quiz, but you may help each other within a group. It can be a good idea to have the paper accessible when you answer the questions, but it is extremely recommended that you read and understand it in advance. Both students must finish the quiz.

C-Assignments

For grade 4, complete the C-assignment below and solve it in groups of ≤ 2 students.

C-1 The Tor design paper refers to Tor as the "second-generation onion router."

> R. Dingledine, N. Mathewson and P. Syverson, Tor: The Second-generation Onion Router, Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13, 2004.

The paper listed below describes what can then be considered the "first-generation onion router".

> D. M. Goldschlag, M. G. Reed, and P. F. Syverson. Hiding routing information. In *Information Hiding*, pp. 137–150, Springer-Verlag, 1996.

Using these papers, identify how the two generations differ from each other. To aid you in your work, in the Tor paper, focus on Sections 1-4 and 6, and in the first generation onion routing paper, focus on the first 3 sections.
Assessment:

- There will be one Moodle-question with statements about differences. Four of these statements are correct and your task is to identify these. You will have 1 hour to finish it.

  You will receive 1 point for each correct answer and -1 point for each error. You will need 3 points to pass. Thus, your best bet is to either pick 3 choices that you are absolutely certain are correct, or pick 5 choices among which you know that you have picked the 4 correct ones.

  It can be a good idea to have the papers accessible when you answer the questions, but it is extremely recommended that you read and understand them in advance. Both students must finish the quiz.