CS 352 **Project 3 DNS client with Authentication server** Due date **December 10,** 2018

For Project 3, you will extend your implementation of Project 2 by changing the root server to be an **authentication** server that authenticates **one** of two TLDS servers that store the mapping between host name and IP addresses. The DNS client program uses a key and a challenge string to create a digest and sends the challenge string as well as the digest to the authentication server. The authentication sever, sends ONLY the challenge string to both the TLDS servers and the TLDS servers return the respective digests. The TLDS servers each store a key and only of them matches the key that the client used to create the digest. Hence, only one of the responses from the TLDS servers match the digest sent by the client. The Authentication server (root server) sends the hostname of the TLDS server with the correct match to the client. The DNS client then connects to that TLDS server with a query string and obtains the A record (if found) from the authenticated TLDS server.

The two TS servers each maintain a DNS_table consisting of three fields: Hostname, IP address, Flag (A). In addition, the TS servers each maintain a key which is used to create a digest from a challenge . When the DNS client connects to the authenticated TLDS server, it sends the hostname as a string. The TLDS server does a look up in the DNS_table and if there is a match, sends the DNS table entry as a string ["Hostname IPaddress A"]. If the host name does not exist then an error string Error: HOST NOT FOUND is returned. Note, that in this Project, the DNS client connects to the TLDS server to get the IP address for a given hostname.

The hostname string, the Key and the challenge will be given one per line in a file (PROJ3-HNS.txt) and the keys one per line will be in a file PROJ3-KEY1.txt and PROJ3-KEY2.txt. The DNS tables entries will also be one per line and will be in PROJ3-TLDS1.txt and PROJ3-TLDS2.txt. The TLDS servers, in addition to reading the DNS entries, will each obtain its key by reading the value from the corresponding key files. TLDS1 will read the key from PROJ3-KEY1.txt and TLDS2 will read the key from PROJ3-KEY2.txt Your client program should output the results to a file RESOLVED.txt. As part of your submission, you need to submit four program files as well as the output file.

The DNS client will read from PROJ3-HNS.txt . Each line will contain a key, a challenge and a hostname (k3521 arianna www.princeton.edu). Using the key and the challenge, the DNS client creates a digest. Then connects to the AS server and sends **ONLY** the challenge string and the digest to the AS server. The AS server then sends **ONLY** the challenge string to both the TLDS servers and collects the responses (digests). Then it does a digest compare and sends the name of the TLDS server that returns the correct digest. Note: as the keys read by the TLDS servers are different, only one digest will match. The DNS client then sends the hostname part of the query string to that TLDS server returned by the AS sever and writes the response received from the TLDS server and the hostname of the TLDS server to the output file RESOLVED.txt

Here is a program snippet that can be used to generate a digest, from a given key string and a challenge string as well as to compare two digests to see if they are equal.

```
import hmac
c1="arianna"
c2="grande"
d1=hmac.new("k3521".encode(),c1.encode("utf-8"))
d2=hmac.new("k3522".encode(),c1.encode("utf-8"))
d3=hmac.new("k3521".encode(),c1.encode("utf-8"))
d4=hmac.new("k3521".encode(),c2.encode("utf-8"))

print(d1.hexdigest()==d2.hexdigest())
print(d1.hexdigest()==d3.hexdigest())
print(d1.hexdigest()==d4.hexdigest())
exit()
```

As always, first start the two TLDS servers, then the AS server and then the client program. Figure out how you will communicate the port number and hostname of TS servers to the AS server and hostname of the AS server to the client program. Choose a suitable data structure for the DNS table as well as to store KEYS and digests.

Here is a sample input line:

K123 obama www.princeton.edu

Here is a sample string sent from the client to the AS server

obama c8c5230843ec953882c5724701e62cc8

Here is a sample string sent from the AS server to the TLDS server

obama

Here is a sample response string sent from the TLDS server back to the AS server

c8c5230843ec953882c5724701e62cc8

Here is a sample output line:

TLDS1 ge.com 68.4.3.2 A

A brief sketch of the interaction among the programs is as shown.