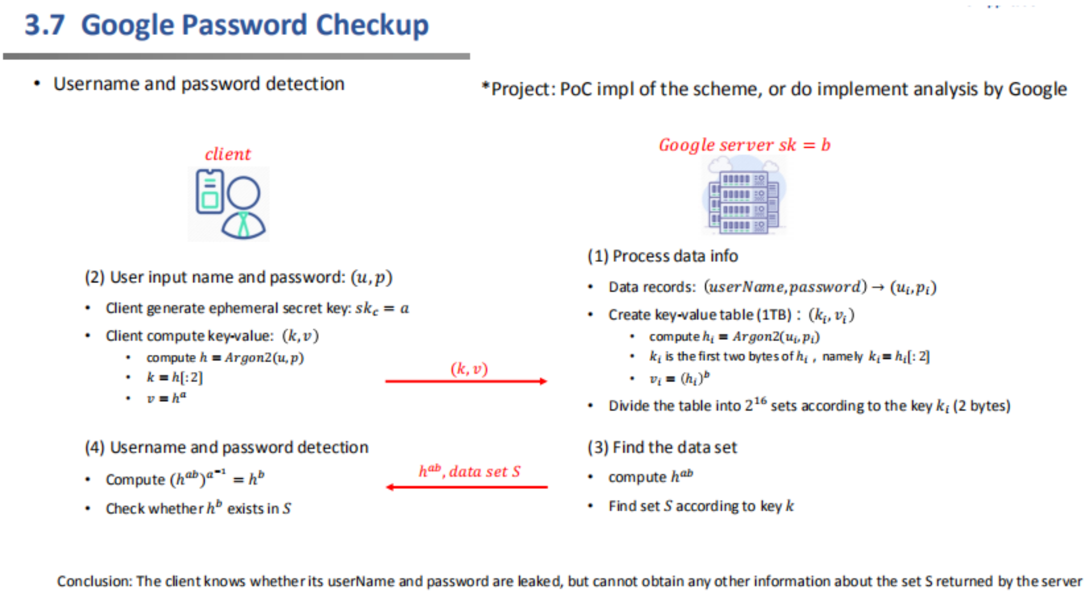
Report

本项目实现了一个简单的PoC scheme，通过网络通信，模拟了server和client的交换，因为只是一个朴素的实现，所以在这里并没有很大的数据量，仅使用了100条数据，每10条数据分成一组，一共10组，并且采用一种最简单的递补法来将hash值映射到椭圆曲线上。在本项目中，使用的hash为sha-256，使用的加密方案为Gmssl库中的sm2。原理如下所示：



3.7 Google Password Checkup

- Username and password detection

*Project: PoC impl of the scheme, or do implement analysis by Google

(1) Process data info

- Data records: $(userName, password) \rightarrow (u_i, p_i)$
- Create key-value table (1TB)： (k_i, v_i)
    - compute $h_i = Argon2(u_i, p_i)$
    - k_i is the first two bytes of h_i , namely $k_i = h_i[:2]$
    - $v_i = (h_i)^b$
- Divide the table into 2^{16} sets according to the key k_i (2 bytes)

(2) User input name and password: (u, p)

- Client generate ephemeral secret key: $sk_c = a$
- Client compute key-value: (k, v)
    - compute $h = Argon2(u, p)$
    - $k = h[:2]$
    - $v = h^a$

(3) Find the data set

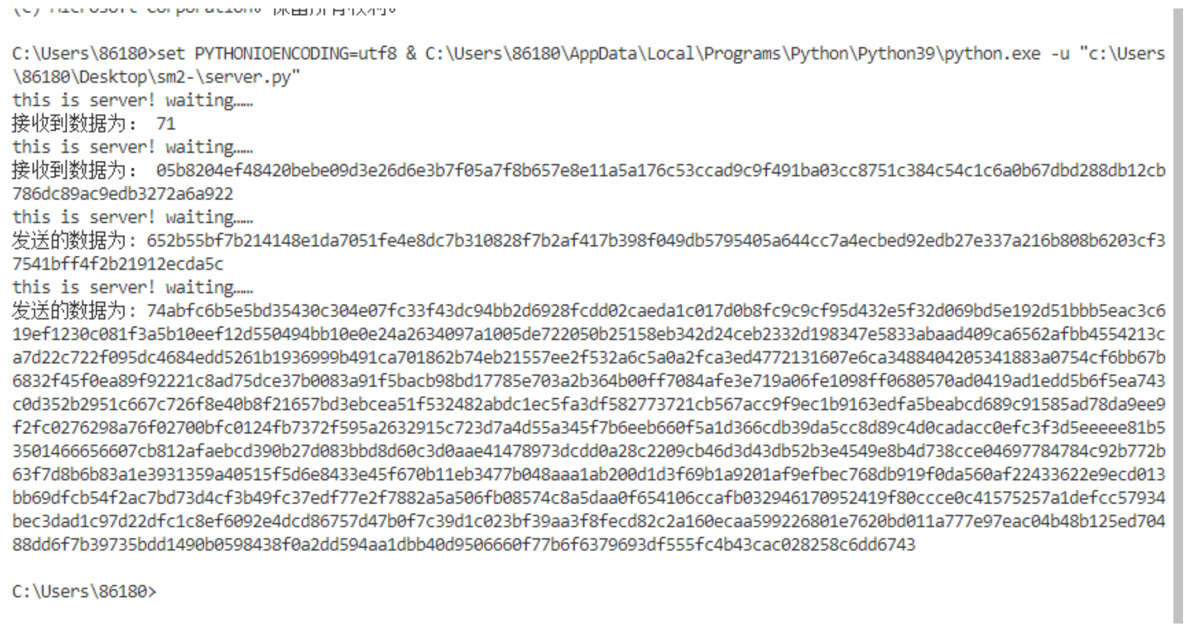
- compute h^{ab}
- Find set S according to key k

(4) Username and password detection

- Compute $(h^{ab})^{a^{-1}} = h^b$
- Check whether h^b exists in S

Conclusion: The client knows whether its userName and password are leaked, but cannot obtain any other information about the set S returned by the server

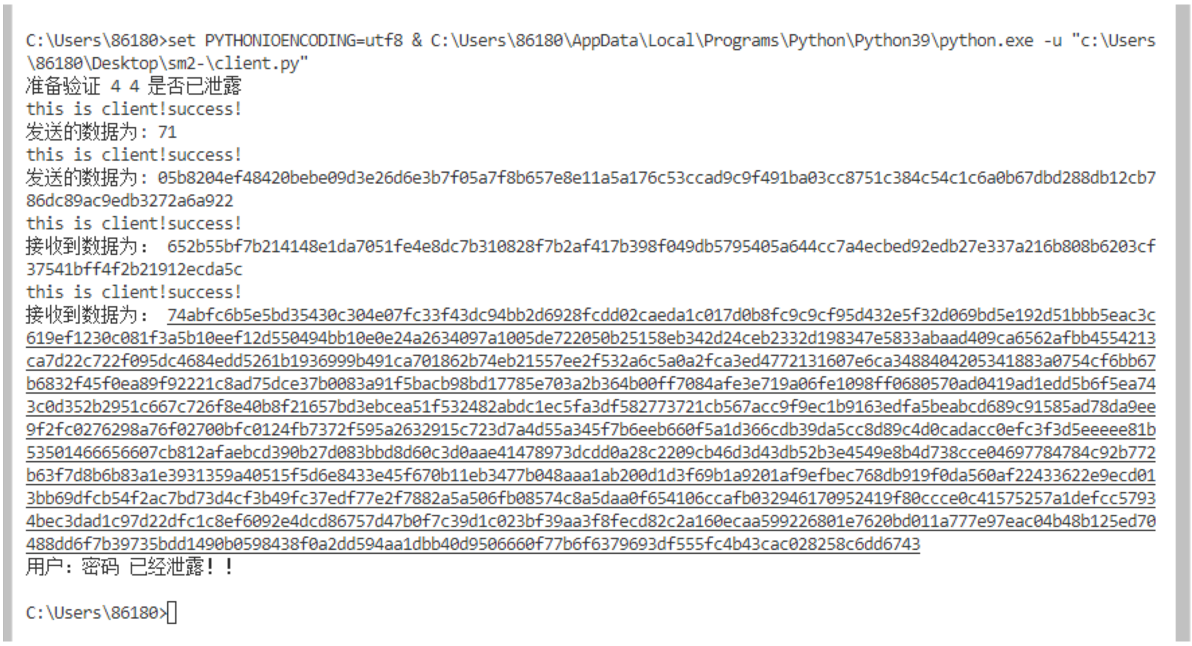
实验结果如下所示，可以看到，对于客户端输入的”用户名：密码“为”4：4“的组合，因为我们在服务端存储的格式为"str(i)：str(i)"，所以应该会显示泄露，成功完成策略复现。



```
(c) Microsoft Corporation。保留所有权利。

C:\Users\86180>set PYTHONIOENCODING=utf8 & C:\Users\86180\AppData\Local\Programs\Python\Python39\python.exe -u "c:\Users
\86180\Desktop\sm2-\server.py"
this is server! waiting.....
接收到数据为：  71
this is server! waiting.....
接收到数据为：  05b8204ef48420bebe09d3e26d6e3b7f05a7f8b657e8e11a5a176c53ccad9c9f491ba03cc8751c384c54c1c6a0b67dbd288db12cb
786dc89ac9edb3272a6a922
this is server! waiting.....
发送的数据为：652b55bf7b214148e1da7051fe4e8dc7b310828f7b2af417b398f049db5795405a644cc7a4ecbed92edb27e337a216b808b6203cf3
7541bff4f2b21912ecda5c
this is server! waiting.....
发送的数据为：74abfc6b5e5bd35430c304e07fc33f43dc94bb2d6928fcdd02caeda1c017d0b8fc9c9cf95d432e5f32d069bd5e192d51bbb5eac3c6
19ef1230c081f3a5b10eef12d550494bb10e0e24a2634097a1005de722050b25158eb342d24ceb2332d198347e5833abaad409ca6562afbb4554213c
a7d22c722f095dc4684edd5261b1936999b491ca701862b74eb21557ee2f532a6c5a0a2fca3ed4772131607e6ca3488404205341883a0754cf6bb67b
6832f45f0ea89f92221c8ad75dce37b0083a91f5bacb98bd17785e703a2b364b00ff7084afe3e719a06fe1098ff0680570ad0419ad1edd5b6f5ea743
c0d352b2951c667c726f8e40b8f21657bd3ebcea51f532482abdc1ec5fa3df582773721cb567acc9f9ec1b9163edfa5beabcd689c91585ad78da9ee9
f2fc0276298a76f02700bfc0124fb7372f595a2632915c723d7a4d55a345f7b6eeb660f5a1d366cdb39da5cc8d89c4d0cadacc0efc3f3d5eeeee81b5
3501466656607cb812afaebcd390b27d083bbd8d60c3d0aae41478973dcdd0a28c2209cb46d3d43db52b3e4549e8b4d738cce04697784784c92b772b
63f7d8b6b83a1e3931359a40515f5d6e8433e45f670b11eb3477b048aaa1ab200d1d3f69b1a9201af9efbec768db919f0da560af22433622e9ecd013
bb69dfcb54f2ac7bd73d4cf3b49fc37edf77e2f7882a5a506fb08574c8a5daa0f654106ccafb032946170952419f80ccce0c41575257a1defcc57934
bec3dad1c97d22dfc1c8ef6092e4dcd86757d47b0f7c39d1c023bf39aa3f8fecd82c2a160ecaa599226801e7620bd011a777e97eac04b48b125ed704
88dd6f7b39735bdd1490b0598438f0a2dd594aa1dbb40d9506660f77b6f6379693df555fc4b43cac028258c6dd6743

C:\Users\86180>
```



```
C:\Users\86180>set PYTHONIOENCODING=utf8 & C:\Users\86180\AppData\Local\Programs\Python\Python39\python.exe -u "c:\Users
\86180\Desktop\sm2-\client.py"
准备验证 4 4 是否已泄露
this is client!success!
发送的数据为: 71
this is client!success!
发送的数据为: 05b8204ef48420bebe09d3e26d6e3b7f05a7f8b657e8e11a5a176c53ccad9c9f491ba03cc8751c384c54c1c6a0b67dbd288db12cb7
86dc89ac9edb3272a6a922
this is client!success!
接收到数据为:  652b55bf7b214148e1da7051fe4e8dc7b310828f7b2af417b398f049db5795405a644cc7a4ecbed92edb27e337a216b808b6203cf
37541bff4f2b21912ecda5c
this is client!success!
接收到数据为:  74abfc6b5e5bd35430c304e07fc33f43dc94bb2d6928fcdd02caeda1c017d0b8fc9c9cf95d432e5f32d069bd5e192d51bbb5eac3c
619ef1230c081f3a5b10eef12d550494bb10e0e24a2634097a1005de722050b25158eb342d24ceb2332d198347e5833abaad409ca6562afbb4554213
ca7d22c722f095dc4684edd5261b1936999b491ca701862b74eb21557ee2f532a6c5a0a2fca3ed4772131607e6ca3488404205341883a0754cf6bb67
b6832f45f0ea89f92221c8ad75dce37b0083a91f5bacb98bd17785e703a2b364b00ff7084afe3e719a06fe1098ff0680570ad0419ad1edd5b6f5ea74
3c0d352b2951c667c726f8e40b8f21657bd3ebcea51f532482abdc1ec5fa3df582773721cb567acc9f9ec1b9163edfa5beabcd689c91585ad78da9ee
9f2fc0276298a76f02700bfc0124fb7372f595a2632915c723d7a4d55a345f7b6eeb660f5a1d366cdb39da5cc8d89c4d0cadacc0efc3f3d5eeeee81b
53501466656607cb812afaebcd390b27d083bbd8d60c3d0aae41478973dcdd0a28c2209cb46d3d43db52b3e4549e8b4d738cce04697784784c92b772
b63f7d8b6b83a1e3931359a40515f5d6e8433e45f670b11eb3477b048aaa1ab200d1d3f69b1a9201af9efbec768db919f0da560af22433622e9ecd01
3bb69dfcb54f2ac7bd73d4cf3b49fc37edf77e2f7882a5a506fb08574c8a5daa0f654106ccafb032946170952419f80ccce0c41575257a1defcc5793
4bec3dad1c97d22dfc1c8ef6092e4dcd86757d47b0f7c39d1c023bf39aa3f8fecd82c2a160ecaa599226801e7620bd011a777e97eac04b48b125ed70
488dd6f7b39735bdd1490b0598438f0a2dd594aa1dbb40d9506660f77b6f6379693df555fc4b43cac028258c6dd6743
用户：密码 已经泄露！！

C:\Users\86180>
```