# Contents

# 1. Experiment 1

Experiment 1 examined how well users could remember unique passwords generated for different accounts when seven restrictions were imposed. These restrictions were chosen because they conform to common recommendations for good passwords, such as "mixing special characters with numbers and letters" is better than mixing numbers and letters or using letters alone. Users were assigned to conditions in which they generated passwords satisfying the restrictions for either three or five accounts what is shown in 1. These numbers of accounts were chosen because they fall within the range of 3–6 accounts held by most users.

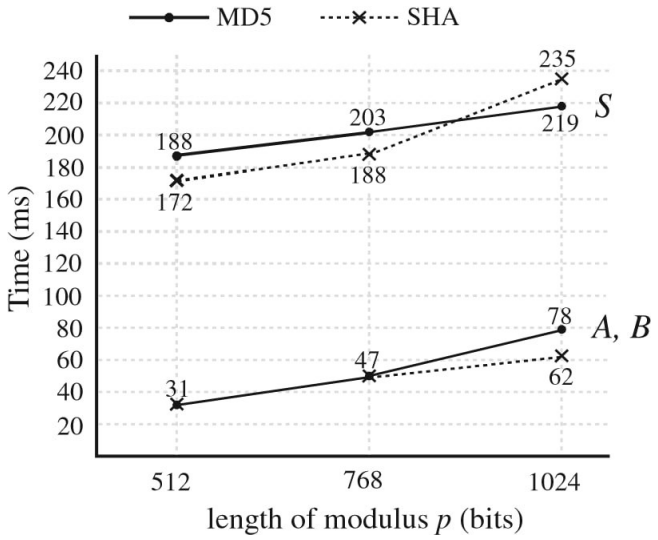


**Figure 1:** Execution time (ms) for 3PAKE with 512 bits, 768 bits and 1024 bits using MD5 and SHA-1 algorithms

## 1.1. Method

### 1.1.1. Participants

Thirty-two students from Purdue University participated for partial credit toward their Introductory Psychology course. All were experienced computer users and were familiar with generating passwords.

### 1.1.2. Apparatus

A program, written in Java, was used to present instructions to participants, record and check the generated passwords, and record the time to generate an acceptable password. Recall time was also measured after a short retention interval of 5min and a long retention interval of 1 week. Generation and recall times were measured by using the Java code that gives the system time in milliseconds, subtracting the time when the password prompt was presented from the time when the password was entered.

$$Pr\left[S_1\right] + Pr\left[S_2\right] \leq \frac{q_{se}}{N} + q_h \cdot \sum_{i=1}^{n} q_i q_{ex} \cdot Success_G^{CDH}\left(t\right) + q_h \cdot q_i \cdot Advantage_G^{CDH}\left(t'\right) \tag{1}$$

Here, we discuss the security features of our proposed protocol as follows in details:

– Identity protection and user anonymity: For the user U, we use PID instead of ID. By using protected pseudonym identities of users instead of real ones, the malicious attacker cannot get users' real identities. Further, in our scheme, while providing authentication of users, service providing servers cannot know users' real identities either. In this way, our protocol provides user anonymity, which can prevent the leakage of private user identities and server identities to malicious attackers. Updating users' pseudonym identities periodically and dynamically can prevent the malicious attacker linking eavesdropped messages of different sessions from the same user.

– Traceability: doesn't provide traceability, but in our scheme, CS can still extract users' real identities and link them with protected pseudonym identities, while provide the function of anonymity between the user U and the service providing server. ID can be retrieved from received CID in formula. This makes our protocol have the feature of traceability. This is a newly-added function in our proposed protocol different from Li et al.'s protocol what is shown in 2.

– Session key agreement: In order to protect the data communication between the user U and the service providing server, a session key need to be negotiated between them in advance, which can further derive encryption keys and MAC keys. In this paper, we only use the hash function and the XOR operation to design a simple but efficient key agreement scheme. By securely exchanging Ni1 and S can separately compute the common session key as in formula.

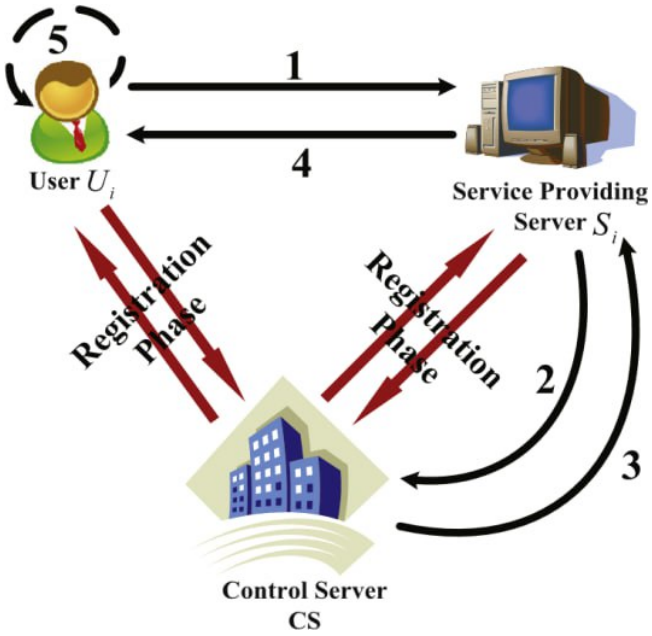


**Figure 2:** Demonstration of registration, authentication and key agreement phases of Li et al.'s protocol

### 1.1.3. Procedure

All participants were tested individually in a quiet, welllit room. Participants were informed that they would be asked to generate passwords for several different "fake" accounts. Sixteen of the participants generated passwords for three accounts, and the other 16 participants did so for five accounts 2. The experiment itself was divided into three parts. In the first part, the participant generated passwords for each account. The experimenter read the instructions for generating the passwords to the participant. For the 3accounts group, the generic account names were: E-mail, bank, and eBay. For the 5-accounts group, the two additional accounts were labeled travel and books.

$$\left|Pr\left[Succ_{v0}\right]-Pr\left[Succ_{q_{akc}}\right]\right| \leq \sum_{i=1}^{q_{ake}} \left|Pr\left[Succ_{v_{i-1}}\right]-Pr\left[Succ_{v_i}\right]\right| \tag{2}$$

Seven password restrictions were imposed, namely that the password:

1. Be at least six characters.
2. Contain an uppercase letter.
3. Contain a lowercase letter.
4. Contain a digit.
5. Contain a special character.
6. Be unique from the passwords generated for the other accounts.
7. Not contain the person's username or any variant of it.

Participants were informed that they would be asked to recall the password for each account after generating all of them. Each participant was asked to enter a username for his/her file and afterwards was presented with a prompt to enter a password for one account. All of the password restrictions remained visible on the screen during this process. If the generated password met all conditions, the prompt for the next account was displayed. If the password did not meet one or more of the restrictions, a prompt to reenter the password was presented along with a list of the restrictions that were not met by the previous password entry. The generation time and number of attempts were recorded and sent directly to a log file. Participants were not given feedback about their generation times. Once the passwords for all accounts were generated and accepted, they were printed on the screen along with their corresponding account names for the participant to review. Participants were not allowed to write the passwords down.
The computer screen was cleared of any information relating to the first part of the experiment, and all participants took a 5-min break. During this time, they left the room and were encouraged to walk around and engage in activities such as getting a drink of water. At the end of this break, the second part of the experiment was started. In this part, participants were presented with a list of the account names for which they had generated passwords. They were informed that one of the account names would appear on the screen, and that they were to recall and enter the password for that account. The participants logged into each account four times in random order. They were also told that they would have a maximum of 10 attempts for each account occurrence. For each occurrence, the login time and number of incorrect attempts were recorded. Before leaving the experiment, the participants were instructed not to write down the passwords that they had generated during the study.
For the last part of the study, participants came back a week later to recall the passwords. The procedure was identical to the second part of the experiment. At the end of the experiment, the log files were compiled, and mean generation and login times for each condition for each participant were computed.

$$Success_G^{CDH}\left(t\right) = Pr\left[S_2\right] \cdot n_2 \cdot b_2 \geq Pr\left[S_2\right] \frac{1}{q_{se}} \frac{1}{q_h} \tag{3}$$

## 1.2. Result

### 1.2.1. Generation time and number of attempts

There were 16 participants assigned to each group. A one-way ANOVA, with group (three or five accounts) as a between-subjects factor, was conducted on two dependent measures: mean generation time and number of attempts 1.
There was no significant difference in generation times or number of attempts to come up with an acceptable password for the 3- and 5-account groups. On average, participants took 29s to generate a password for an account and arrived at an acceptable password after a single attempt. The number of characters for each password ranged from 6 to 15, with the mean length of each password being 9.1 characters for the three accounts group and 8.5 characters for the five accounts group.

**Table 1:** Security functionality comparison of our protocol and two other related protocols.

| *Security functionality* | *Our proposed protocol* | *Li et al.'s protocol* | *Sood et al.'s protocol* |
|---|---|---|---|
| Identity protection and user anonymity smart card forgery attack | Yes | Yes | Yes |
| Dynamic identity updating | Yes | Yes | No |
| Traceability | Yes | Yes | Yes |
| Mutual authentication | Yes | Yes | Yes |
| Session key agreement | Yes | Yes | Yes |
| Password updating | Yes | Yes | Yes |
| Resistance of insider attack and smart card forgery attack | Yes | No | No |
| Resistance of stolen smart card attack | Yes | Yes | No |
| Resistance of replay attack | Yes | No | No |
| Resistance of Denial-of-Service attack | Yes | No | No |
| Resistance of eavesdropping attack | Yes | No | No |
| Resistance of masquerade attack | Yes | No | No |

### 1.2.2. Login time and number of attempts

Mean login time and number of attempts for each participant were submitted to 2 (Group: three or five accounts)2 (Recall delay: 5-min or 1 week) repeatedmeasures ANOVAs with recall delay as a within-subjects factor and group as a between-subjects factor. There were no significant effects for login time. However, the main effect of number of accounts was significant for the mean number of login attempts. Participants in the 5-accounts made more errors in recalling the passwords than those in the 3-accounts condition. There was no significant effect of recall delay, and it did not significantly interact with the number of accounts.

$$\overline{X} = \sum_{i=1}^{n} \frac{ln\,|X|}{H_i\,(ID_A, ID_B, ID_S, g^i)^{pw}\,A} = g^{ez} \tag{4}$$

where $X = g(mod\ p)$
$A = X \cdot M$

We also examined the number of participants who forgot the password for one or more accounts. For the 3-accounts group, one participant failed to recall a password during short-term recall, and

only two participants were unable to recall the password for one account during long-term recall. In contrast, for the 5-accounts group, two participants did not recall the passwords for three of the five accounts, both at short and long-term recall. In addition, five participants were not able to remember a password for one account during short-term recall, and three participants for one account and one participant for two accounts at long-term recall.

### 1.2.3. Crackability of passwords

We limited the password cracking time for which the lc5 program was run to approximately 4h. The lc5 program was able to crack a significantly larger percentage of the passwords from the 3-accounts group than from the 5-accounts group. The requirement to generate five unique passwords rather than just three apparently forced participants to be more creative in their password generation.