Hindawi
Complexity
Volume 2019, Article ID 2875676, 8 pages
https://doi.org/10.1155/2019/2875676

*Research Article*

# Polynomial-Based Google Map Graphical Password System against Shoulder-Surfing Attacks in Cloud Environment

**Zhili Zhou**,[1] **Ching-Nung Yang**,[2] **Yimin Yang**,[3] **and Xingming Sun**[1]

[1]*Jiangsu Engineering Center of Network Monitoring, School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, China*
[2]*Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan*
[3]*Department of Computer Science, Lakehead University, Thunder Bay, Canada*

Correspondence should be addressed to Zhili Zhou; zhou_zhili@163.com and Ching-Nung Yang; cnyang@gms.ndhu.edu.tw

Received 14 September 2019; Revised 18 October 2019; Accepted 25 October 2019; Published 16 November 2019

Guest Editor: Yuan Yuan



Text password systems are commonly used for identity authentication to access different kinds of data resources or services in cloud environment. However, in the text password systems, the main issue is that it is very hard for users to remember long random alphanumeric strings due to the long-term memory limitation of the human brain. To address this issue, graphical passwords are accordingly proposed based on the fact that humans have better memory for images than alphanumeric strings. Recently, a Google map graphical password (GMGP) system is proposed, in which a specific location of Google Map is preset as a password for authentication. Unfortunately, the use of graphical passwords increases the risk of exposing passwords under shoulder-surfing attacks. A snooper can easily look over someone's shoulder to get the information of a location on map than a text password from a distance, and thus the shoulder-surfing attacks are more serious for graphical passwords than for text passwords. To overcome this issue, we design a polynomial-based Google map graphical password (P-GMGP) system. The proposed P-GMGP system can not only resist the shoulder-surfing attacks effectively, but also need much fewer challenge-response rounds than the GMGP system for authentication. Moreover, the P-GMGP system is extended to allow a user to be authenticated in cloud environment effectively and efficiently.

## 1. Introduction

In modern digital life, people cannot avoid to use passwords for identity authentication. Recently, the emerging technologies such as cloud computing [1–3] and big data processing [4–12] develop very rapidly. Under this background, the password systems play more and more important roles to allow legal users to access different kinds of data resource or service in cloud environment. Most traditional password authentication systems use the combination of alphanumeric strings such as numbers, symbols, and mixed-case letters as passwords, i.e., text passwords. To resist brute force or random guessing attacks, users usually adopt long random alphanumeric strings as strong text passwords, but these strong passwords are hard to remember due to the long-term memory limitations of the human brain. Thus, many users tend to choose simple passwords that are easy to remember, but they will be easily cracked by malicious attackers. Therefore, it is quite difficult to achieve a good trade-off between passwords' strength and memorability in text password authentication systems.

Due to the fact that humans have better memory for images over alphanumeric strings, the graphical password systems have been proposed accordingly to address the issues of text password systems. In the literature, there are two well-known kinds of graphical password systems, i.e., pass-points graphical password (PPGP) systems [13, 15–18] and cued click-points graphical password (CCPGP) systems [19–25]. In the PPGP systems, users click a sequence of preset points on a picture as password for the registration and login processes, as shown in Figure 1. Actually, it is harder to remember a set of points than a single point on a picture. Therefore, instead of using only one picture, the CCPGP systems have been proposed to use a series of pictures and users only need to click one point on each picture. After clicking a point on a picture, the next picture will appear according to the point clicked on the previous picture, as shown in Figure 2.

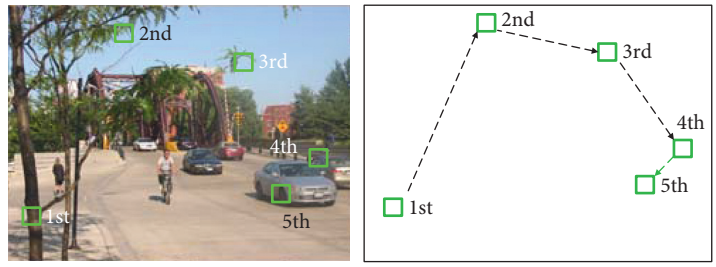


Figure 1: Selection of a sequence of points on a single picture in the PPGP system [13]. The numbers shown on the image mean the orders of clicks.

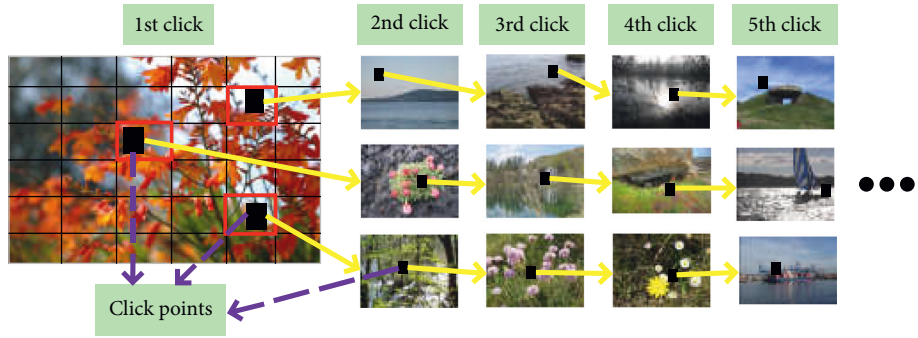


Figure 2: Selection of a sequence of points on a picture set in the CCPGP system [14].

However, CCPGP systems need to store a large number of pictures in the database. Recently, Spitzer et al. [26] integrated application programming interface (API) of Google Map into CCPGP to propose a Google map graphical password (GMGP) system. Because Spitzer et al.'s GMGP system uses online Google Map services, it does not need additional storage space for storing a large number of pictures.

However, both text passwords and graphical passwords are compromised by shoulder-surfing attacks. The shoulder-surfing attacks will be more serious for graphical passwords than for text passwords since a snooper can easily get the information of a location of map than a text password from a distance. In this work, we design a polynomial-based Google map graphical password (P-GMGP) system against shoulder-surfing attacks. The proposed P-GMGP system not only resists the shoulder-surfing attacks effectively, but also significantly reduces time complexity of authentication compared to Spitzer et al.'s GMGP system.

In cloud environment, a user usually needs to access different kinds of services or data resources from multiple servers. The proposed P-GMGP system can be easily extended to allow a user to be authenticated by $M$ servers simultaneously. Thus, the extended P-GMGP system can be applied successfully in cloud environment.

The rest of this paper is organized as follows. Section 2 introduces the previous work. The motivation and contributions are introduced in Section 3. The proposed P-GMGP system and the extended P-GMGP system are described in Section 4. Conclusions are drawn in Section 5.

## 2. Previous Work

Spitzer et al.'s GMGP [26] adopted Google Map API into CCPGP, so that it does not require a huge storage space for storing pictures. Instead of receiving a natural picture, users will receive a Google Map from the server in each layer. Consider a seven-layer Spitzer et al.'s GMGP, where a Google Map in each layer is subdivided into 256 grids. Seven layers provide $256^7 = 2^{56}$ possible choices, and thus the GMGP system has the same security strength as a DES algorithm with a 56-bit key. To pass authentication, users should correctly click a set of grids on Google Maps. Because a user has the knowledge of the password point, he can easily select a specific grid that includes this password point. After clicking the selected grid on the map, Spitzer et al.'s GMGP will zoom out (enlarge) this grid and display this enlarged Google Map on next layer. By selecting the correct grids in all seven layers (note: every grid in each layer should include the password point), users can pass the authentication successfully. Figure 3 illustrates the login/authentication process of the GMGP system.

Denote the longitude and latitude of password point $P$ as $(x_P, y_P)$ and the longitude and latitude of response point $R_i$ as $(x_i, y_i)$ in the $i$th layer, where $1 \le i \le n$ and $(x_i, y_i)$ belong to the grid $G_i \in [1, 256]$. The following briefly describes registration and login/authentication phases for an $n$-layer Spitzer et al.'s GMGP system.

*2.1. Registration Phase.* In this phase, a user ($U$) with their identifier $ID_U$ applies for a login account to the verification server ($V$). To achieve secure communication, the password

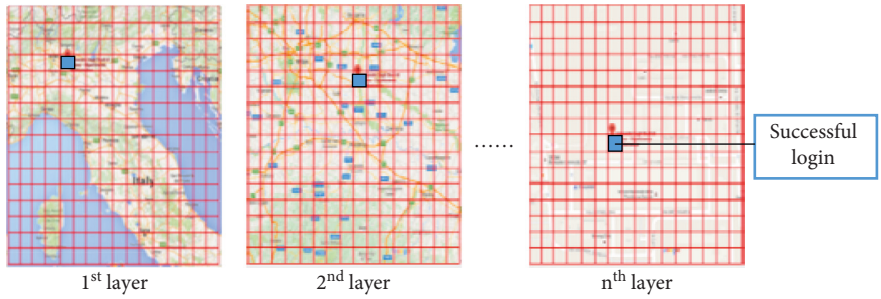


Figure 3: The login process by selecting a grid on each layer of Google Map in the GMGP system [26].

points and the response points are sent via a Hypertext Transfer Protocol Secure (HTTPS) channel.

*Step (1).* $U$ chooses a password point $P = (x_P, y_P)$ which is easy to remember

*Step (2).* $U$ sends the registration request $\{ID_U, P\}$ to $V$ via a secure channel

*Step (3).* $V$ stores the user's ID and the longitude and latitude of the password point $\{\mathrm{ID}_U, P\}$ in the database

*2.2. Login/Authentication Phase.* In this phase, $U$ tries to login to $V$ and $V$ verifies $U$.

*Step (1).* $U$ starts login by sending his identifier $\{\mathrm{ID}_U\}$ to $V$.

*Step (2).* After receiving $\{\mathrm{ID}_U\}$, $V$ generates an arbitrarily large Google Map $M_i$ including $P$, where $1 \le i \le n$, and then sends $\{M_i\}$ to $U$. If $1 \le i \le n-1$, repeat steps (3-1)~(3-3).

*Step (3-1).* After receiving the map $\{M_i\}$, by the knowledge about $P$, $U$ chooses a response point $R_i = (x_i, y_i)$ in the grid $G_i$ where the point $P$ is located. Then, the user sends $\{R_i\}$ to $V$.

*Step (3-2).* After receiving $\{R_i\}$, $V$ determines $G_i$ by $R_i$ and $M_i$.

*Step (3-3).* $V$ zooms out the map in grid $G_i$ to generate the next map $M_{i+1}$, and then sends $\{M_{i+1}\}$ to $U$.

*Step (3-4).* For the last layer (i.e., $i = n$), $U$ chooses a response point $R_n = (x_n, y_n)$ in the grid $G_n$ where the point $P$ is located, and then sends $\{R_n\}$ to $V$.

*Step (4).* $V$ verifies whether every grid in the corresponding layer includes the password point, i.e., $P \overset{?}{\in} G_i$ for $1 \le i \le n$.

*Step (5).* If all the verifications pass, the login/authentication is successful.

## 3. Motivation and Contributions

In Spitzer et al.'s GMGP, the response point is sent by the plaintext type via a secure channel, i.e., a HTTPS channel. Generally, HTTPS can be resistant to replay attacks. Thus, users can use the same password point in each login/authentication phase and do not need to consider the replay attacks.

However, using the same password point may lead to the increase of risk of exposing password under shoulder-surfing attacks. In fact, the shoulder-surfing attacks are usually more serious for graphical password than for text password since a snooper can look over someone's shoulder to get the information of a location on map more easily than a text password from a distance. Although the $n$-layer Spitzer et al.'s GMGP system needs $n$ challenge-response rounds in login/authentication phase, there is an easy way to steal login information just from the last round. The reasons are given below.

Because the map $M_{i+1}$ is enlarged from the grid $G_i$ in the previous map $M_i$ and thus $M_{i+1} = G_i$, the grids have the following property, i.e., $G_{i+1} \subset G_i$, $1 \le i \le n-1$, as shown in the following equation:

$$G_{i+1} \subset M_{i+1} \longrightarrow G_{i+1} \subset G_i \quad (\because M_{i+1} = G_i, 1 \le i \le n-1). \tag{1}$$

Consequently, a snooper could capture the last response point $R_n = (x_n, y_n)$ by a direct observation, and thus he knows the grid $G_n$. By the property $G_{i+1} \subset G_i$ and the knowledge of $G_n$, $1 \le i \le n-1$, attackers can know the correct grids to send the corresponding response points in all layers to pass the authentication.

To resist the shoulder-surfing attacks, the user should not directly expose any knowledge of the password point when they try to login to the server. To this end, motivated by the theory of zero-knowledge proof, we propose a polynomial-based Google map graphical one-time password (P-GMGP) system. In this system, instead of directly clicking the password point, the user clicks a point on the straight line (i.e., one-degree polynomial) generated by connecting the password point and a challenge point to pass the authentication. Since the attackers hardly know any information of the password point from the clicked point in the user's login phase, it is possible for the proposed system to resist the shoulder-surfing attacks. Moreover, the password point used in the proposed system is a one-time password, which can further improve the security.

In addition, another flaw of Spitzer et al.'s GMGP is that the $n$-layer Spitzer et al.'s GMGP system needs $n$ challenge-response rounds for authentication since the user needs to click a correct point on each of $n$ maps in different levels. Thus, the authentication process is time-consuming especially for large $n$. On the contrary, in the

proposed P-GMGP system, the user only needs to click one point on a line, and thus the system can reduce $n$ challenge-response rounds to a single round to speed up the authentication.

We also consider the application scenarios of cloud environment. Suppose that a cloud service provider builds a multiserver system, and these servers may collaborate to provide various services or data resources to the clients. When a user needs to login to multiple servers (say $M$ servers) to get services or data resources, the user should login $M$ times in Spitzer et al.'s GMGP system. On the contrary, the proposed P-GMGP system can be easily extended to allow the user to be authenticated by $M$ servers, simultaneously, which will significantly reduce the authentication cost compared to Spitzer et al.'s GMGP system. Therefore, the extended P-GMGP system is suitable for cloud environment.

## 4. The Proposed P-GMGP and Extended P-GMGP Systems

*4.1. P-GMGP.* Our design concept is based on one-degree polynomial (i.e., a straight line). The proposed P-GMGP system can resist shoulder-surfing attacks effectively. The registration phase of the P-GMGP system is the same as that of Spitzer et al.'s GMGP system. Thus, we only show the login/authentication phase. The operation by $U$ and the operation with the help of computing device (say smartphone $S$) are separated to more clearly understand the process.

*4.1.1. Login/Authentication Phase.* In this phase, $V$ sends a challenge point $C = (x_C, y_C)$ and $U$ responds a response point $R$ for login.

*Step (1).* $U$ starts login by sending his identifier $\{\mathrm{ID}_U\}$ to $V$.

*Step (2).* After receiving $\{\mathrm{ID}_U\}$, $V$ randomly selects a challenge point $C$ and uses current timestamp $T$, and then sends $\{C, T\}$ to $U$.

*Step (3).* By receiving $\{C, T\}$, $S$ determines a straight line $L$ from two points, $C$ and the shifted password point $P' = (x_{P'}, y_{P'})$, where $x_{P'} = H(x_P, T)$ and $y_{P'} = H(y_P, T)$, and shows this line $L$ on screen.

*Step (4).* $U$ randomly selects a response point $R = (x_R, y_R)$ on the line $L$ and sends $R$ to $V$.

*Step (5).* $V$ verifies whether $R$ is on $L$ or not (note: because $V$ has the information of $C$, $P$, and $T$, it can determine the line $L$). If the verification passes, the authentication is successful.

As we know, any two points will determine a straight line (i.e., one-degree polynomial), while a straight line contains infinite points theoretically. The proposed P-GMGP is based on the following Lemma.

**Lemma 1.** *Suppose that any three distinct points $P_1$, $P_2$, and $P_3$ in a plane are on a straight line (one-degree polynomial). For the straight line drawn by connecting $P_1$ and $P_2$, one knowing the point $P_3$ can verify whether $P_3$ is on the line. If others do not have information of $P_3$, they cannot know $P_3$ from the line.*

*Proof.* The two points $P_1$ and $P_2$ can determine a line represented by a one-degree polynomial $f(x) = a_0 + a_1 x$. One can use the coordinates of $P_3 = (x_3, y_3)$ to determine whether $P_3$ is on the line by checking $y_3 \stackrel{?}{=} a_0 + a_1 x_3$. On the other hand, because a straight line contains an infinite set of points theoretically, others cannot know the point $P_3$ from the straight line $f(x) = a_0 + a_1 x$.

**Theorem 1.** *The proposed P-GMGP is a one-time password scheme. Also, attackers cannot get the password point from the intercepted point, challenge point, and response point.*

*Proof.* Denote the challenge point, the response point, the shifted password point, and the line as $C_i$, $R_i$, $P_i'$, and $L_i$ for the current timestamp $T_i$, respectively. For the different timestamps $T_i$ and $T_j$, where $T_i \neq T_j$, we have different lines.

(i.e., $L_i \neq L_j$). Thus, the response points are not static. From Step (2)~Step (4), the points $C_i$, $R_i$, and $P_i'$ are on the straight line $L_i$. From Lemma 1, we can conclude that $V$ can verify whether $R_i$ is on the line, and meanwhile attackers cannot obtain information about $P_i'$ from the line. As shown in Figure 4, the intercepted point of $L_i$ and $L_j$ is just a normal point on $L_i$ and $L_j$ rather than the password point $P_i'$. Moreover, because a straight line theoretically has an infinite set of points, we cannot reveal any information of $P_i'$ and $P_i'$ from the intercepted point. Even though attackers obtain $C_i$ and $R_i$, where $1 \le i \le n$, from the above description, they have no information of $P_i' = (x_{P_i'}, y_{P_i'})$. Also, the coordinates of password point are protected by hash function, i.e., $x_{P_i'} = H(x_P, T_i)$ and $y_{P_i'} = H(y_P, T_i)$, attackers cannot get the original password point $P = (x_P, y_P)$.

Basically, the security of our P-GMGP is based on two factors: (i) infinite points theoretically contained in a line, which does not expose the shifted password points and (ii) the one-way property of cryptographic hash functions $H(\cdot)$, which could resist severe collision attacks and preimage attacks [27]. By factor (i), attackers cannot guess the shifted password points from the lines. Even though attackers have the shifted password points (note: it is impossible), they still have no information of the original password point due to the one-way hash function in factor (ii).

It is worth noting that, since it is hard for some users to exactly click points on a line in practice, we set a tolerant distance TD = 100 m to improve the usability of the proposed P-GMGP system, so that the user can easily pass the authentication by clicking any point within the tolerant distance TD = 100 m to the line. Moreover, although an infinite number of points are contained in a line theoretically, that is not true in a line drawn on Google Map. Therefore, in practice, the probability that attackers can successfully guess the shifted password points from the line can be computed by

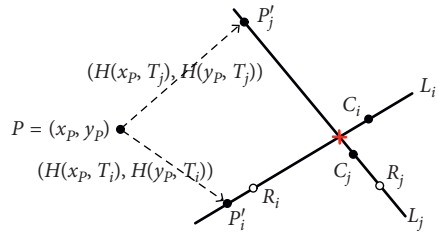


Figure 4: Diagrammatical representation of password point, shifted password points, and challenge and response points for different timestamps in the P-GMGP system.

$$P_G \approx \frac{\text{Area}_R}{\text{Area}_P} = \frac{\pi \times \text{TD}^2}{L \times 2\text{TD}}, \quad (2)$$

where $\text{Area}_R$ means the area of the region on which the user can click any point to successfully pass the authentication and $\text{Area}_P$ means the area of the possible region in which the response points are located in Google Map. As the tolerant distance TD = 100 m is used in the proposed P-GMGP system and the length of line drawn on the whole Google Map is usually very large, by equation (2), the probability $P_G$ is equal to a very small value. That means it is still very hard for attackers to successfully guess the shifted password points from the line.

Another advantage of P-GMGP is to speed up the authentication process. When using an $n$-layer GMGP system, a user should select $n$ response points in $n$ challenge-response rounds in the login/authentication phase. As shown in Step (4) in the proposed P-GMGP system, the user only needs to send one response point in a single round.

*4.2. Extended P-GMGP.* Also, we consider how to efficiently login to $M$ servers (say $V_1, V_2, \ldots, V_M$) to get services or data resources in cloud environment. By using $M$-degree polynomial instead of one-degree polynomial, we can easily extend the P-GMGP system to allow a user to be authenticated by $M$ servers, simultaneously. The registration process of the extended P-GMGP system is the same as that of the P-GMGP system. After registration, users have $M$ password points $P_{V_m} = (x_{P_{V_m}}, y_{P_{V_m}})$ for the $M$ servers $V_m$, $1 \le m \le M$, respectively. The modified login/authentication phase is described below.

*4.2.1. Login/Authentication Phase.* In this phase, a representative server of these $M$ servers sends a challenge point $C$ and $U$ selects a response point $R$ for the login.

*Step (1).* $U$ starts login process by sending the login request $\{\text{ID}_U, \text{ID}_{V_1}, \text{ID}_{V_2}, \ldots, \text{ID}_{V_M}\}$ to a representative server (say $V_1$). Note that $V_1$ has to collect all shifted password points from the other verification server $P'_{V_m} = (H(x_{P_{V_m}}, T), H(y_{P_{V_m}}, T))$, $2 \le m \le M$.

*Step (2).* After receiving $\{\text{ID}_U, \text{ID}_{V_1}, \text{ID}_{V_2}, \ldots, \text{ID}_{V_M}\}$, $V_1$ randomly selects a challenge point $C = (x_C, y_C)$ and uses the current timestamp $T$, and then sends $\{C, T\}$ to $U$.

*Step (3).* By receiving $\{C, T\}$, $S$ determines a curve line $\Omega$ ($M$-degree polynomial) from $(M+1)$ points, $C$ and $M$ shifted password points $P'_{V_m} = (H(x_{P_{V_m}}, T), H(y_{P_{V_m}}, T))$, $1 \le m \le M$. Then, it shows this curve line $\Omega$ on screen.

*Step (4).* $U$ selects a response point $R = (x_R, y_R)$ on $\Omega$ and sends $R$ to $V_1$.

*Step (5).* $V_1$ verifies whether $R$ is on $\Omega$ or not (Note: because $V_1$ has $C$ and all shifted password points, it can determine $\Omega$). If the verification passes, the authentication is successful.

The following lemma and theorem for the extended P-GMGP are similar to Lemma 1 and Theorem 1 for P-GMGP, respectively. By using the same argument, they can be easily proved.

**Lemma 2.** *Suppose that any $(M+2)$ distinct points $P_1$, $P_2, \ldots$, and $P_{M+2}$ in a plane are on a curve line $\Omega$ ($M$-degree polynomial). By public $(M+1)$ points (say $P_1 \sim P_{M+1}$), one knowing the point $P_{M+2}$ can easily verify whether $P_{M+2}$ is on the curve line $\Omega$. If one does not have the information of $P_{M+2}$, they cannot know the point $P_{M+2}$ from the curve line $\Omega$.*

**Theorem 2.** *The extended P-GMGP is a one-time password scheme. Attackers cannot get the password point from intercepted points and challenge and response points. Also, servers can verify whether the user has $M$ passwords simultaneously $P_{V_m} = (x_{P_{V_m}}, y_{P_{V_m}})$, $1 \le m \le M$.*

For the case $M = 2$, Figure 5 shows the password points, shifted password points, and challenge and response points for different timestamps in the extended P-GMGP system. Like the P-GMGP system, the security is also based on the two factors mentioned above. The password points $P_{V_1}$ and $P_{V_2}$ are not on the parabolic curves $\Omega_i$ and $\Omega_j$; according to factor (ii), the password points are shifted and protected by hash function. Also, attackers cannot know the information of the shifted password points $P'_{i,V_1}$, $P'_{i,V_2}$, $P'_{j,V_1}$, and $P'_{j,V_2}$; according to factor (i), infinite points are theoretically contained in a line. Thus, the extended P-GMGP system has high security. Additionally, the servers can verify whether the user has $M$ passwords $P_{V_m} = (x_{P_{V_m}}, y_{P_{V_m}})$, $1 \le m \le M$, simultaneously, by checking if $R$ is on $\Omega$ or not.

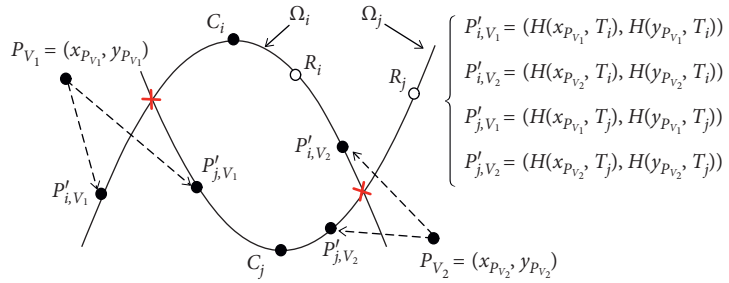


Figure 5: Diagrammatical representation of password point and challenge and response points for different timestamps in the extended P-GMGP when $M = 2$.

### 4.3. Discussion

*4.3.1. Entering Response Point with the Help of Smartphone.* In Spitzer et al.'s GMGP system, with the help of Google Map functions (locating, sizing, zooming, and panning via Google Maps API), users can personally select a response point according to the knowledge about password point. In the proposed P-GMGP system, after receiving the challenge point, we need the smartphone $S$ to generate the $M$-degree curve (note: users can check whether the curve generated by smartphone is correct or not because he knows the password points). Then, he chooses any one point on the line as a response point.

*4.3.2. Compatibility of the P-GMGP System and the Extended P-GMGP System.* Actually, the two password schemes are compatible. Users can login to multiple servers one by one. Also, they can login to any set of multiple servers, simultaneously. The login/authentication processes of the two systems are very similar, and the only difference is that users choose a response point on a straight line (for $M = 1$) and a curve line (for $M \geq 2$).

*4.3.3. Single-Round Authentication.* An $n$-layer Spitzer et al.'s GMGP system needs $n$ challenge-response rounds for authentication. The proposed P-GMGP system only sends a response point on a straight line in a single round, and this will speed up the authentication process. By the extended P-GMGP system, even authenticating by $M$ servers, the user still needs to send one point on a curve line in one round.

*4.3.4. Comparison among Different Graphical Password Systems.* We also make comparison among the GMGP system, P-GMGP system, and extended P-GMGP system, which is summarized in Table 1. From this table, the proposed two systems, i.e., the proposed P-GMGP system and its extended version can resist the shoulder-surfing attacks effectively, while the CCPGP and GMGP systems cannot. Moreover, the proposed two systems only need a challenge-response round in login phase, which is much less than the CCPGP and GMGP systems. The extended GMGP system can allow a user to login to multiple servers, simultaneously, while the others cannot. Finally, since the CCPGP system needs to store a large number of pictures for authentication, it has high storage load. On the contrary, the GMGP, P-GMGP, and extended P-GMGP systems use the online Google Maps, and thus they have no storage load for storing pictures.

Table 1: The comparison among different graphical password systems.

| | CCPGP | GMGP | P-GMGP | Extended |
|---|---|---|---|---|
| Resistance to the shoulder-surfing attacks | No | No | Yes | Yes |
| Number of challenge-response rounds | n | n | 1 | 1 |
| Login to multiple servers, simultaneously? | No | No | No | Yes |
| Storage load | Large number of pictures | No need | No need | No need |

## 5. Conclusion

The shoulder-surfing attacks are more serious for graphical passwords than for text passwords, since a snooper can easily get the information of a location on map than a text password from a distance. Moreover, using the same password in the GMGP system may increase the risk of exposing the password under shoulder-surfing attacks. In this work, we have presented the polynomial-based graphical password systems, i.e., the P-GMGP system and its extended version.

In the P-GMGP system and the extended P-GMGP system, since users do not need to directly click the password points to pass the authentication, the two systems can resist shoulder-surfing attacks effectively. The security of P-GMGP and extended P-GMGP is based on the two mentioned factors, i.e., the property of line and one-way hash function, and thus the two systems have high security. Also, the user only needs to send one response point in one round for login, and thus the proposed systems need much less authentication time than the GMGP system. In addition, since the extended P-GMGP system allows the user to be authenticated by $M$ servers, simultaneously, it is very suitable for multiserver environment.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by MOST under contracts 108-2634-F-259-001 through Pervasive Artificial Intelligence Research (PAIR) Labs, Taiwan, in part by the National Natural Science Foundation of China under Grant nos. 61972205, 61602253, U1836208, U1536206, U1836110, and 61672294, in part by the National Key R&D Program of China under Grant no. 2018YFB1003205, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, and in part by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China.

## References

[1] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, and X. Xu, "A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems," *World Wide Web*, vol. 5, pp. 1–23, 2019.

[2] L. Qi, S. Meng, X. Zhang et al., "An exception handling approach for privacy-preserving service recommendation failure in a cloud environment," *Sensors*, vol. 18, no. 7, pp. 1–11, 2018.

[3] L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, and J. Chen, "A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment," *Future Generation Computer Systems*, vol. 88, pp. 636–643, 2018.

[4] X. Wang, W. Wang, L. T. Yang, S. Liao, D. Yin, and M. J. Deen, "A distributed HOSVD method with its incremental computation for big data in cyber-physical-social systems," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 481–492, 2018.

[5] Z. Zhou, J. Q. M. Wu, and X. Sun, "Multiple distances-based coding: toward scalable feature matching for large-scale web image search," *IEEE Transactions on Big Data*, 2019.

[6] L. Qi, Q. He, F. Chen et al., "Finding all you need: web APIs recommendation in web of things through keywords search," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1063–1072, 2019.

[7] X. Wang, L. T. Yang, H. Li, M. Lin, J. Han, and B. O. Apduhan, "NQA," *ACM Transactions on Embedded Computing Systems*, vol. 18, no. 4, pp. 1–21, 2019.

[8] H. Liu, H. Kou, C. Yan, and L. Qi, "Link prediction in paper citation network to construct paper correlation graph," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 233, 2019.

[9] Z. Zhou, Y. Mu, and Q. M. J. Wu, "Coverless image steganography using partial-duplicate image retrieval," *Soft Computing*, vol. 23, no. 13, pp. 4927–4938, 2019.

[10] L. Qi, W. Dou, W. Wang, G. Li, H. Yu, and S. Wan, "Dynamic mobile crowdsourcing selection for electricity load forecasting," *IEEE Access*, vol. 6, pp. 46926–46937, 2018.

[11] X. Zhang, W. Dou, Q. He et al., "A generic framework for fast tree isolation based ensemble anomaly analysis," in *Proceedings of the IEEE 33rd International Conference on Data Engineering*, pp. 983–994, San Diego, CA, USA, May 2017.

[12] X. Wang, L. T. Yang, Y. Wang, X. Liu, Q. Zhang, and M. J. Deen, "A distributed tensor-train decomposition method for cyber-physical-social services," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 4, pp. 1–15, 2019.

[13] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "PassPoints: design and longitudinal evaluation of a graphical password system," *International Journal of Human-Computer Studies*, vol. 63, no. 1-2, pp. 102–127, 2005.

[14] A. M. Iranna and P. Patil, "Graphical password authentication using persuasive cued click point," *International Journal of Advanced Research in Electrical Electronics & Instrumentation Engineering*, vol. 2, no. 7, pp. 142–144, 2013.

[15] S. Wiedenbeck, J. Waters, J. C. Birget, A. Brodskiy, and N. Memon, "Authentication using graphical passwords: effects of tolerance and image choice," in *Proceedings of the Symposium on Usable Privacy and Security*, pp. 1–12, Pittsburgh, PA, USA, July 2005.

[16] P. C. V. Oorschot, A. Salehi-Abari, and J. Thorpe, "Purely automated attacks on PassPoints-style graphical passwords," *IEEE Transactions on Information Forensics & Security*, vol. 5, no. 3, pp. 393–405, 2010.

[17] A. E. Dirik, N. Memon, and J.-C. Bimerget, "Modeling user choice in the PassPoints graphical password sche," in *Proceedings of the Symposium on Usable Privacy and Security*, pp. 20–28, Pittsburgh, PA, USA, July 2007.

[18] A. Meiappane, V. P. Venkataesan, and V. Premanand, "Security enhancement in shoulder surfing attacks using passpoints for random similar images (PRSIm)," *International Journal of Computer Networks and Applications*, vol. 2, no. 2, pp. 84–91, 2015.

[19] H. M. Sun, S. T. Chen, J. H. Yeh, and C. Y. Cheng, "A shoulder surfing resistant graphical authentication system," *IEEE Transactions on Dependable & Secure Computing*, vol. 15, no. 2, pp. 180–193, 2016.

[20] S. Chiasson, P. C. V. Oorschot, and R. Biddle, "Graphical password authentication using cued click points," in *Proceedings of the European Symposium on Research in Computer Security*, pp. 359–374, Dresden, Germany, September 2007.

[21] S. Chiasson, A. Forget, R. Biddle, and P. C. Van Oorschot, "User interface design affects security: patterns in click-based graphical passwords," *International Journal of Information Security*, vol. 8, no. 6, pp. 387–398, 2009.

[22] X. Liu, J. Qiu, L. Ma, H. Gao, and Z. Ren, "A novel cued-recall graphical password scheme," in *Proceedings of the International Conference on Image and Graphics*, pp. 949–956, Hefei, China, August 2011.

[23] S. Chiasson, E. Stobert, A. Forget, R. Biddle, and P. C. Van Oorschot, "Persuasive cued click-points: design, implementation, and evaluation of a knowledge-based authentication mechanism," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 2, pp. 222–235, 2011.

[24] S. Hande, N. Dighade, R. Bhusari, M. Shende, and P. H. Agrawal, "Image based authentication for folder security using persuasive cued click-points and SHA," *IOSR Journal of Computer Engineering*, vol. 16, no. 2, pp. 124–128, 2014.

[25] H. M. Aljahdali and R. Poet, "Challenge set designs and user guidelines for usable and secured recognition-based graphical passwords," in *Proceedings of the IEEE 13th International Conference on Trust, Security and Privacy in Computing and*

*Communications*, pp. 973–982, Beijing, China, September 2014.

[26] J. Spitzer, C. Singh, and D. Schweitzer, "A security class project in graphical passwords," *Journal of Computing Sciences in Colleges*, vol. 26, no. 2, pp. 7–13, 2010.

[27] J.-P. Aumasson, W. Meier, and F. Mendel, "Preimage attacks on 3-pass HAVAL and step-reduced MD5," in *Proceedings of the International Workshop on Selected Areas in Cryptography*, pp. 120–135, Cincinnati, OH, USA, October 2008.