Weaknesses of Password Authentication Scheme Based on Geometric Hashing

Martin Stanek
Department of Computer Science, Comenius University
Mlynská Dolina, 842 48 Bratislava, Slovak
(Email: stanek@dcs.fmph.uniba.sk)

(Received May 9, 2014; revised and accepted Jan. 16 & May 5, 2015)

Abstract

We show that a recently proposed password authentication scheme based on geometric hashing has several security weaknesses, and that the use of this scheme should be avoided in practice.

Keywords: Authentication, geometric hashing, password

1 Introduction

Password authentication schemes are an important class of cryptographic protocols. They provide a mechanism that allows a user to authenticate to a server by using his/her password. Various protocols were proposed, e.g. [6, 9, 10, 12, 13, 14, 15], and many attacks on such schemes appeared, e.g. [2, 4, 5, 7, 8, 11, 13]. A survey on password authentication schemes can be found in [16].

Recently, a new password authentication scheme was proposed in [17]. The scheme is based on geometric hashing. The authors claim the security of the scheme, namely a resilience against replay attack, off-line guessing, stolen-verifier, denial of service, and man-in-the-middle attacks.

For brevity, we will call the scheme from [17] "the geometric scheme". We show several weaknesses of the geometric scheme:

– Contrary to the claim stated in [17] the geometric scheme is not resistant against stolen-verifier attack. We show how an attacker can authenticate using user's stolen verifier. Moreover the verifier can also be used to recover user's identity and password.

– The performance of the geometric scheme can be easily degraded by a malicious user. First, the attacker can cause a significant grow of the internal data structure (a hash table) used for storing users' verifiers. Second, the attacker is able to cause a large number of collisions in the hash table, thus increasing the complexity of lookups performed during authentication.

– The scheme allows a substantial reduction of potential identity/password pairs.

The paper is organized as follows. We briefly introduce the geometric scheme in Section 2. The weaknesses are presented in Section 3.

2 The Geometric Scheme

We present the geometric scheme in sufficient detail to make our paper self-contained and to allow the reader to understand the findings.

2.1 Preliminaries

The geometric scheme is based on problem of geometric hashing [3]; for more recent work on bucketing problems, see [1].

Let S be the set of points in two-dimensional space $\mathbb{R}^2$: $S = \{(x_i, y_i)\}_{i=1}^{n}$. Let $\theta \in \langle 0, \pi)$ be an angle of projection – it uniquely determines the line going through the origin point $(0, 0)$ such that the angle between the line and x-axis is θ. A projection of point $s = (x, y)$ at angle θ is defined as a value $|s|_\theta = x\cos\theta + y\sin\theta$. For given S and θ the following notions are defined:

– The span of the projection:

$$\mathrm{span}_\theta(S) = \max\{||s_i|_\theta - |s_j|_\theta|\,;\ s_i, s_j \in S\}.$$

– The resolution of the projection:

$$\mathrm{res}_\theta(S) = \min\{||s_i|_\theta - |s_j|_\theta|\,;\ s_i, s_j \in S \text{ and } i \neq j\}.$$

– The length of the projection:

$$\mathrm{len}_\theta(S) = \mathrm{span}_\theta(S)/\,\mathrm{res}_\theta(S).$$

In order to have the smallest projection value starting at zero, the value C is computed as $C = \min\{|s|_\theta;\ s \in S\}$. Then an adjusted projection is a shifted version of the projection: $P_\theta(s) = |s|_\theta - C$.

2.2 Initial Configuration

User's identity id and password pw represent a point $(\mathsf{id}, \mathsf{pw}) \in \mathbb{R}^2$. The server computes an angle θ^* that minimizes the length of the projection for initial set of n users S_U, e.g. using an $O(n^2 \log n)$ algorithm from [3]. The idea is to create even-sized buckets and place each $(\mathsf{id}, \mathsf{pw})$ pair into a corresponding bucket according its adjusted projection. Let C^* be the value calculated for shifting the projection, and let res_{θ^*} be the resolution of the projection computed for S_U and θ^*.

The server builds a hash table with entries $(\mathrm{idx}_{\theta^*}(\mathsf{id}, \mathsf{pw}), P_{\theta^*}(\mathsf{id}, \mathsf{pw}))$, where the index value $\mathrm{idx}_{\theta^*}(\mathsf{id}, \mathsf{pw}) = \lfloor P_{\theta^*}(\mathsf{id}, \mathsf{pw}) / \mathrm{res}_{\theta^*} \rfloor$ serves as a key for the hash table. Possible collision are treated in a standard way by linked list of colliding values.

The length of the projection is minimized to make the size of the hash table reasonable small. The use of adjusted projection ensures that the hash table index starts at zero. The server publishes the values θ^*, C^* and res^*_θ computed for the initial set of users, and stores the hash table. Note that the server stores neither passwords, nor user identities in clear.

2.3 Authentication

Authentication involves two parties – client (user) and server. The protocol consists of a single message prepared by the client and sent to the server, and a verification performed by the server.

The client knows his/her own identity id, password pw, and public values published by the server – θ^*, C^* and res^*_θ. Let H be a cryptographic hash function, and let t_c be a fresh time stamp. The client computes values $m_1 = P_{\theta^*}(\mathsf{id}, \mathsf{pw})$, $m_2 = H(\mathsf{pw} \oplus \mathsf{id})$, $M_1 = \mathrm{idx}_{\theta^*}(\mathsf{id}, \mathsf{pw})$, $M_2 = H(m_1 \oplus t_c)$, $M_3 = \mathsf{pw} \oplus m_1$, and $M_4 = P_{\theta^*}(m_2, \mathsf{pw})$. The client sends to the server the 5-tuple: M_1, M_2, M_3, M_4, t_c.

The server performs the following verification steps after receiving the values M_1, M_2, M_3, M_4, t_c:

1) If time stamp t_c is not recent enough or M_1 if out range of the hash table, the server rejects the authentication.

2) Index M_1 is used to get all $P_{\theta^*}(\mathsf{id}', \mathsf{pw}')$ values from the hash table that share this index. The sever searches through these values for $m_1' = P_{\theta^*}(\mathsf{id}', \mathsf{pw}')$ such that $H(m_1' \oplus t_c) = M_2$. If the m_1' is not found, the server rejects.

3) The server extracts $\mathsf{pw}' = M_3 \oplus m_1'$ and $\mathsf{id}' = (m_1' - \mathsf{pw} \cdot \sin\theta^* + C) / \cos\theta^*$. Finally, the server computes $m_2' = H(\mathsf{pw}' \oplus \mathsf{id}')$ and verifies that $P_{\theta^*}(m_2', \mathsf{pw}') = M_4$. The authentication is accepted only after successful verification.

3 Attacks and Weaknesses

We discuss several attacks and weaknesses of the proposed geometric hashing in this section.

Stolen-verifier attack - Authentication.

The stolen-verifier attack covers a situation when the attacker has stolen a user's verifier stored in the server. The attacker then tries to authenticate as the user. The scheme should be resistant to such attack. The verifier in the geometric scheme is the value $P_{\theta^*}(\mathsf{id}, \mathsf{pw})$. Contrary to the claim stated in [17], the geometric scheme is not resistant to the stolen-verifier attack.

First, observe that M_1, M_3 and M_4 messages are constant in each authentication for given $(\mathsf{id}, \mathsf{pw})$ pair. The only part of the authentication that depends on the time stamp t_c is $M_2 = H(m_1 \oplus t_c) = H(P_{\theta^*}(\mathsf{id}, \mathsf{pw}) \oplus t_c)$. Thus, the attacker can forge the authentication of the user:

1) Generate a fresh time stamp t_c.
2) Compute $M_2 = H(P_{\theta^*}(\mathsf{id}, \mathsf{pw}) \oplus t_c)$ using the stolen-verifier $P_{\theta^*}(\mathsf{id}, \mathsf{pw})$.
3) Send M_1, M_2, M_3, M_4, t_c, where M_1, M_3 and M_4 are eavesdropped messages from some previous run of the authentication.

Stolen-verifier attack - Identity and password recovery.

Even more damaging is the observation that the verifier $P_{\theta^*}(\mathsf{id}, \mathsf{pw})$ (i.e. m_1) is used for password encryption. Therefore, a possession of the verifier and an eavesdropped authentication messages allow the attacker to recover the password $\mathsf{pw} = M_3 \oplus m_1$. Having pw, the identity id can be computed as well.

Hash table overflow and underflow.

Since the angle θ^* is not adjusted when users change their passwords or new users are added, the angle and the corresponding resolution can become suboptimal. The parameters θ^*, C^* and res^*_θ are public. Therefore a malicious user (attacker) can cause:

1) An overflow of hash table. The authors of the geometric scheme acknowledge the possibility that the hash table can stretch. Seemingly, they do not recognize the possibility to cause an extreme increase of the hash table size. It is easy to maximize the index of $(\mathsf{id}, \mathsf{pw})$ pair by maximizing the value of formula $\mathsf{id} \cdot \cos\theta^* + \mathsf{pw} \cdot \sin\theta^*$.
2) An underflow of hash table. Similarly to the overflow problem, minimizing $\mathsf{id} \cdot \cos\theta^* + \mathsf{pw} \cdot \sin\theta^*$ cause, possibly large, negative index value. The hash table then must by adjusted for these indices and enlarged correspondingly.

Collisions in the hash table.

The values θ^*, C^* and res^*_θ are public. Hence anybody can compute index $\mathrm{idx}_{\theta^*}(\mathsf{id}, \mathsf{pw})$ for any pair (id, pw). Moreover, for given id, one can easily compute various pw such that $\mathrm{idx}_{\theta^*}(\mathsf{id}, \mathsf{pw})$ is some predetermined value. Thus, the attacker (or a colluding set of users) can register a large number of users (or change their passwords) such that the idx_{θ^*} values collide. This causes long chains to store P_{θ^*} values and substantially degrade the performance of the hash table.

In addition, the index for the user is send as a part of his/her authentication in M_1 message. This allows to target any particular user and cause performance degradation just for this index.

Reducing the identity/password space.
The message M_1 contains the index value for the user. The attacker can use M_1 for reducing the space of possible identity/password pairs. If $\mathrm{idx}_{\theta^*}(\mathsf{id}', \mathsf{pw}') \neq M_1$ the attacker can conclude that $(\mathsf{id}', \mathsf{pw}')$ is not valid for the user. When the attacker learns user's id by some other means, it allows to test the password. Assuming uniform distribution of identities/passwords the test reduces the possible space by factor n, where n denotes the number of buckets in the hash table, i.e. the maximal index. Usually n increases with increasing the user-base. Hence, the larger the user-base, the more selective test. Moreover, combining with the weakness of hash table overflow, the attacker can choose the factor of the reduction.

Representation of values over real numbers.
The paper [17] lacks any detail how user's identity and password, commonly given as strings of printable characters, are mapped into domain of real numbers to perform geometric hashing. Moreover, some rounding or truncating should be applied for operations like recovering user's id in the verification procedure. The precise formula will depend on the mapping.

4 Conclusion

We showed the insecurity of the password authentication scheme based on geometric hashing proposed in [17]. The attacks and weaknesses presented in the this paper suggest that the use of the scheme should be avoided in practice. Moreover, because of numerous issues it is probably not worth to fix the scheme – it would be a completely new authentication scheme.

Acknowledgment

The author acknowledges support by VEGA 1/0259/13.

References

[1] P. K. Agarwal, B. K. Bhattacharya, and S. Sen, "Output-sensitive algorithms for uniform partitions of points," *Proceedings of 10th International Symposium on ISAAC*, LNCS 1741, pp. 403–414, Springer, 1999.

[2] C. M. Chen and W. C. Ku, "Stolen-verifier attack on two new strong-password authentication protocols," *IEICE Transactions on Communications*, vol. E85-B, no. 11, pp. 2519–2521, 2002.

[3] D. Comer and M. J. O'Donnell, "Geometric problems with application to hashing," *SIAM Journal on Computing*, vol. 11, no. 2, pp. 217–226, 1982.

[4] T. H. Feng, C. H. Ling, and M. S. Hwang, "Cryptanalysis of Tan's Improvement on a Password Authentication Scheme for Multi-server Environments," *International Journal of Network Security*, vol. 16, no. 4, pp. 318–321, 2014.

[5] D. He, J. Chen, and J. Hu, "Weaknesses of a remote user password authentication scheme using smart card," *International Journal of Network Security*, vol. 13, no. 1, pp. 58–60, 2011.

[6] M. S. Hwang, C. C. Lee, and Y. L. Tang, "A simple remote user authentication scheme," *Mathematical and Computer Modeling*, vol. 36, no. 1-2, pp. 103–107, 2002.

[7] M. Kim and Ç. K. Koç, "A Simple Attack on a Recently Introduced Hash-Based Strong-Password Authentication Scheme," *International Journal of Network Security*, vol. 1, no. 2, pp. 77–80, 2005.

[8] W. C. Ku, C. M. Chen, and H. L. Lee, "Weaknesses of Lee-Li-Hwangs hash-based password authentication scheme," *ACM Operating Systems Review*, vol. 37, no. 4, pp. 19–25, 2003.

[9] W. C. Ku, "A hash-based strong password authentication scheme without using smart cards," *ACM Operating System Review*, vol. 38, no. 1, pp. 29–34, 2004.

[10] C. C. Lee, L. H. Li, and M. S. Hwang, *A remote user authentication scheme using hash functions*, *ACM Operating System Review*, vol. 36, no. 4, pp. 23–29, 2002.

[11] C. C. Lee, C. H. Liu, and M. S. Hwang, "Guessing attacks on strong-password authentication protocol," *International Journal of Network Security*, vol. 15, no. 1, pp. 64–67, 2013.

[12] I. E. Liao, C. C. Lee, and M. S. Hwang, "A password authentication scheme over insecure networks," *Journal of Computer and System Sciences*, vol. 72, no. 4, pp. 727–740, 2006.

[13] C. Lin, H. Sun, and T. Hwang, "Attacks and solutions on strong-password authentication," *IEICE Transactions on Communications*, vol. E84-B, no. 9, pp. 2622–2627, 2001.

[14] M. Peyravian and N. Zunic, "Methods for protecting password transmission," *Computers and Security*, vol. 19, no. 5, pp. 466–469, 2000.

[15] M. Sandirigama, A. Shimizu, and M. T. Noda, "Simple and secure password authentication protocol (SAS)," *IEICE Transactions on Communications*, vol. E83-B, no. 6, pp. 1363–1365, 2000.

[16] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password Authentication Schemes: Current Status and Key Issues," *International Journal of Network Security*, vol. 3, no. 2, pp. 101–115, 2006.

[17] X. Zhuang, C. C. Chang, Z. H. Wang, and Y. Zhu, "A Simple Password Authentication Scheme Based on Geometric Hashing Function," *International Journal of Network Security*, vol. 16, no. 4, pp. 271–277, 2014.

Martin Stanek is an Associate Professor in the Department of Computer Science, Comenius University. He received his PhD. in computer science from Comenius University. His research interests include cryptography and information security.