Password-Based Authenticated Key Exchange

David Pointcheval

ENS, Paris, France*

Abstract. *Authenticated Key Exchange* protocols enable several parties to establish a shared cryptographically strong key over an insecure network using various authentication means, such as strong cryptographic keys or short (*i.e.*, low-entropy) common secrets. The latter example is definitely the most interesting in practice, since no additional device is required, but just a human-memorable password, for authenticating the players.

After the seminal work by Bellovin and Merritt, many settings and security notions have been defined, and many protocols have been proposed, in the two-user setting and in the group setting.

1 Introduction

Key exchange protocols are cryptographic primitives used to provide several users (two or more), communicating over a public unreliable channel, with a secure session key. This thus allows establishment of virtual secure channels over insecure networks, which is one of the main practical applications of cryptography. Bellare and Rogaway gave the first foundations in [13, 14], but password-based authentication required more work: in this setting, where the authentication means is a short secret chosen from a small set of possible values (a four-digit pin, for example), the brute-force method which consists in trying all the possible values in the dictionary succeeds after a rather small number of attempts. This attack is called *on-line dictionary attack* and is unavoidable. But its damages can be limited by a policy that invalidates or blocks the use of a password if a certain number of failed attempts has occurred, unless failures are undetectable [27].

This paper presents a brief survey on Password-based Authenticated Key Exchange (PAKE) protocols, with a presentation of some security models in Section 2, and relations to practice. Section 3 deals with some practical constructions.

2 Security Models

Bellare, Pointcheval and Rogaway [12], and Boyko, MacKenzie and Patel [16] first formalized security of Password-based Authenticated Key Exchange, in two different frameworks.

[*] CNRS – UMR 8548 and INRIA – EPI Cascade.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 390–397, 2012.

2.1 Game-Based Security

The former model [12], the so-called *Find-then-Guess* scenario, is in the indistinguishability-based framework where an adversary should not be able to get an advantage significantly greater than q_S/N (or at most $\mathcal{O}(q_S)/N$ for some technicality reasons) in distinguishing a random session key from a real session key, if q_S is the number of active attacks and N the size of the dictionary. It has thereafter been improved to the *Real-or-Random* scenario [7]. More precisely, the adversary is given access to oracles: Execute-queries model passive attacks, Send-queries model active attacks, Corrupt-queries model corruptions with the leakage of long-term secrets, Reveal-queries model bad uses of session keys and thus the leakage of ephemeral secrets, and Test-queries model the semantic security of the session key with a real or random answer. In the Find-then-Guess scenario, only one Test-query can be asked, whereas in the Real-or-Random scenario many Test-queries can be asked with either always-real or always-random answers. The latter is clearly at least as strong as the former. But while both scenarios were known to be equivalent for encryption schemes [11], a linear loss in the number of Test-queries makes them quite different for PAKE, where the advantage should remain in $\mathcal{O}(q_s)/N$, whatever the number of Test-queries. We have then showed [7] that in this Real-or-Random scenario, Reveal-queries are not useful anymore, hence simplifying the security games.

2.2 Simulation-Based Security

The latter model [16] is in the simulation-based framework, with an ideal functionality in which the adversary is allowed to check one password per session. This models on-line dictionary attacks. Excepted this test instance password, no information is leaked about the passwords and the session keys.

2.3 Universal Composability

In both above models, one formalized the fact that, with an active attack, the adversary can basically test one password, whereas passive eavesdropping does not (computationally) leak any information. The goal is essentially to rule out *off-line dictionary attacks* in which the adversary makes some active and passive attacks, and then makes an off-line brute-force attack on the dictionary. On-line brute-force attacks, which are unavoidable, should be the only possible way to have some information about the session keys, and thus many interactions with a real player are required.

However, there were still some limitations on the password distributions and for composition with other protocols, which were overcome by Canetti, Halevi, Katz, Lindell and MacKenzie [24]. They indeed provided an ideal functionality in the Universally Composable (UC) security framework [23], see Figure 1. This functionality also models on-line dictionary attacks with a TestPwd-query that can be asked once to each user in sessions. An important property is that passwords are chosen by the environment which then hands them to the parties

The functionality $\mathcal{F}_{\mathsf{PAKE}}$ is parameterized by a security parameter k. It interacts with an adversary $\mathcal{S}$ and a set of parties $P_1, \ldots, P_n$ via the following queries:

- P_i asks for a $(\mathsf{\bf NewSession}, \mathsf{\bf sid}, \boldsymbol{P_i}, \boldsymbol{P_j}, \boldsymbol{pw})$: Send $(\mathsf{NewSession}, \mathsf{sid}, P_i, P_j)$ to $\mathcal{S}$. If this is the first NewSession-query, or if this is the second NewSession-query and there is a record (P_j, P_i, pw'), then record (P_i, P_j, pw) and mark this record fresh.
- $\mathcal{S}$ asks for a $(\mathsf{\bf TestPwd}, \mathsf{\bf sid}, \boldsymbol{P_i}, \boldsymbol{pw'})$: If there is a record of the form (P_i, P_j, pw) which is fresh, then do:
  - If $pw = pw'$, mark the record compromised and reply with “correct guess”;
  - If $pw \neq pw'$, mark the record interrupted and reply with “wrong guess”.
- $\mathcal{S}$ asks for a $(\mathsf{\bf NewKey}, \mathsf{\bf sid}, \boldsymbol{P_i}, \boldsymbol{sk})$: If there is a record of the form (P_i, P_j, pw), and this is the first NewKey-query for P_i, then:
  - If this record is compromised, or either P_i or P_j is corrupted, then output (sid, sk) to player P_i;
  - If this record is fresh, and there is a record (P_j, P_i, pw') with $pw' = pw$, and a key sk' was sent to P_j, and (P_j, P_i, pw) was fresh at the time, then output (sid, sk') to P_i;
  - In any other case, pick a new random key sk' of length k and send (sid, sk') to P_i.

  Either way, mark the record (P_i, P_j, pw) as completed.

Fig. 1. The PAKE Ideal Functionality $\mathcal{F}_{\mathsf{PAKE}}$

as inputs. This guarantees security even in the case where two honest players execute the protocol with two different passwords: the environment can emulate any distribution, mistypes of passwords and related passwords. Also note that allowing the environment to choose the passwords guarantees forward secrecy. This functionality mimics quite well some concrete requirements, but still, some leakage of information is not modeled, and could be exploited by a real-life adversary, whereas the ideal functionality does not allow it to the ideal-world adversary.

Explicit Authentication. With the above functionality, if neither party is corrupted, then they both end up with a uniformly-distributed session key, either the same key if the passwords are the same (success), or independent keys if the passwords are different (failure). Furthermore, the adversary learns nothing about the keys and the passwords, and even nothing about the status of the session (success or failure), but the users either. *Explicit authentication*, or mutual authentication modeled in [5], provides the players with a session key if and only if the passwords are the same, informing the adversary of success or not. This is an interesting additional feature, which is also more relevant in practice. In the real life, the adversary anyway learns whether the protocol succeeded or not, since in the latter case the communication stops.

Combined with the split functionality [10], it also allows to remove the TestPwd-query since the NewKey-query would reveal to the adversary whether the passwords are the same or not, by leaking the success or failure status. The split functionality allows the adversary to split a session between users Alice and

Bob into two sessions, one between Alice and the adversary trying to impersonate Bob, and a second one between Bob and the adversary trying to impersonate Alice. When the adversary plays with Alice, in case of success, this means it has guessed Alice's password, which is similar to the TestPwd-query.

Contributiveness. In the $\mathcal{F}_{\mathsf{PAKE}}$ functionality, if one party is corrupted, or if the adversary successfully guessed the player's password, the adversary is granted the right to fully determine the session key. Note that as soon as a party is corrupted, the adversary anyway learns the key, so one can think that nothing is lost by allowing it to fully determine it. But this is precisely the difference between *key agreement* and *key distribution* protocols.

In case of groups, this makes a huge difference. Hence the more recent functionality proposed by Abdalla, Catalano, Chevalier and Pointcheval [4] which provides the *contributiveness* property to Group Password-based Authenticated Key Exchange (GPAKE), see Figure 2. PAKE is a particular case of GPAKE with

The functionality $\mathcal{F}_{\mathsf{GPAKE}}$ is parameterized by a security parameter k, and the parameter t of the contributiveness. It interacts with an adversary $\mathcal{S}$ and a set of parties $P_1, \ldots, P_n$ via the following queries:

- P_i asks for a **$(\mathsf{NewSession}, \mathsf{sid}, \mathsf{Pid}, P_i, pw_i)$**: If this is the first NewSession-query for P_i, where Pid is a set of at least two distinct identities containing P_i, record $(\mathsf{sid}, \mathsf{Pid}, P_i, pw_i)$, mark it fresh, and send $(\mathsf{sid}, \mathsf{Pid}, P_i)$ to $\mathcal{S}$. Ignore any subsequent NewSession-queries with a different Pid set. If all the players involved in Pid have submitted their NewSession-queries, then record $(\mathsf{sid}, \mathsf{Pid}, \mathsf{ready})$ and send it to $\mathcal{S}$.
- $\mathcal{S}$ asks for a **$(\mathsf{TestPwd}, \mathsf{sid}, \mathsf{Pid}, P_i, pw')$**: If there exists a record of the form $(\mathsf{sid}, \mathsf{Pid}, P_i, pw_i)$ which is fresh:
  - If $pw_i = pw'$, mark the record compromised and reply with "correct guess";
  - If $pw_i \neq pw'$, mark the record interrupted and reply with "wrong guess".
- $\mathcal{S}$ asks for a **$(\mathsf{NewKey}, \mathsf{sid}, \mathsf{Pid}, sk)$**: If there is a record of the form $(\mathsf{sid}, \mathsf{Pid}, \mathsf{ready})$, then, denote by n_c the number of corrupted players, and
  - If all $P_i \in \mathsf{Pid}$ have the same passwords and $n_c < t$, choose $sk' \in \{0,1\}^k$ uniformly at random and store $(\mathsf{sid}, \mathsf{Pid}, sk')$.
  - If all $P_i \in \mathsf{Pid}$ have the same passwords but $n_c \geq t$, store $(\mathsf{sid}, \mathsf{Pid}, sk)$.

  In both cases, for all $P_i \in \mathsf{Pid}$, mark the record $(\mathsf{sid}, \mathsf{Pid}, P_i, pw_i)$ completed. In any other case, store $(\mathsf{sid}, \mathsf{Pid}, \mathsf{error})$, and for all $P_i \in \mathsf{Pid}$, mark the record $(\mathsf{sid}, \mathsf{Pid}, P_i, pw_i)$ error. When the key is set, report the result (either error or completed) to $\mathcal{S}$.
- $\mathcal{S}$ asks for a **$(\mathsf{SendKey}, b, \mathsf{sid}, \mathsf{Pid}, P_i)$**: If $P_i \in \mathsf{Pid}$ and there is a recorded tuple $(\mathsf{sid}, \mathsf{Pid}, \alpha)$ where $\alpha \in \{0,1\}^k \cup \{\mathsf{error}\}$, send $(\mathsf{sid}, \mathsf{Pid}, \alpha)$ to P_i if $b = 1$ or $(\mathsf{sid}, \mathsf{Pid}, \mathsf{error})$ if $b = 0$.
- $\mathcal{S}$ asks for a **$(\mathsf{Corrupt}, \mathsf{sid}, \mathsf{Pid}, P_i)$**: If there is a recorded tuple $(\mathsf{sid}, \mathsf{Pid}, P_i, pw_i)$, then reveal pw_i to $\mathcal{S}$. If there also is a recorded tuple $(\mathsf{sid}, \mathsf{Pid}, sk)$, that has not yet been sent to P_i, then send $(\mathsf{sid}, \mathsf{Pid}, sk)$ to $\mathcal{S}$.

Fig. 2. The Contributory GPAKE Ideal Functionality $\mathcal{F}_{\mathsf{GPAKE}}$

groups of size 2. The latter property allows the adversary to fully determine the session key only if it has corrupted enough players, more than a threshold. This threshold can even be maximal: as soon as a player is honest, if a common key is generated, it is uniformly distributed in an unpredictable way. This means that no player has a more important role, and so there is no player to corrupt in priority for the adversary. As explained above, and as done in [5], one can even remove TestPwd-queries, allowing the adversary to split the group into several subgroups, with sub-session-IDs, where the adversary plays the role of the other users.

3 Constructions

3.1 Two-Party Password-Based Authenticated Key Exchange

Bellovin and Merritt [15] proposed the first scheme, the so-called Encrypted Key Exchange (EKE), see Figure 3 for a sketch of the protocol, where $\mathcal{E}$ is assumed to be an encryption scheme onto the group $\mathbb{G}$, sometimes modeled as an ideal cipher. A first security analysis has been provided in the indistinguishability-based framework, in the ideal-cipher model [12], followed by several proofs of variations [18, 19, 8], trying to reduce the need of ideal models but still keeping the initial efficiency of EKE. EKE has also been studied in the simulation-based framework, in the random-oracle model [16], followed by studies in the UC framework [3] with security against adaptive corruptions, but still in ideal models. Our "simple PAKE" protocols [8] are definitely the most efficient, with a random oracle only for extracting the session key, with a security analysis in the Find-then-Guess scenario, under the CDH assumption.

Katz, Ostrovsky and Yung [33] proposed the first practical scheme, but still less efficient than above schemes, in the standard model with a common reference string, followed by a generalization from Gennaro and Lindell (GL) [29, 28], using the power of smooth-projective hash functions [26], in the Find-then-Guess scenario. Many variations [24, 6, 34, 31, 35] have thereafter been proposed, to get security in the UC framework, to improve round efficiency, or to rely on new assumptions.

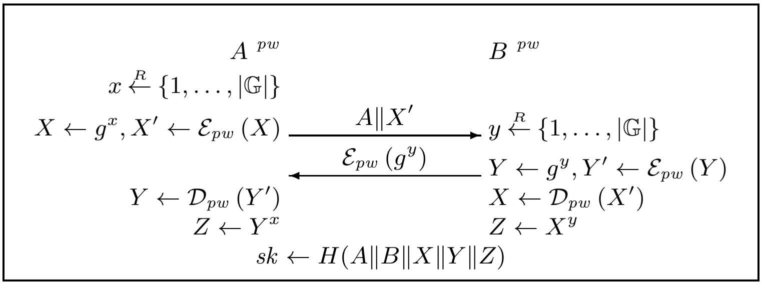
$A^{\ pw}$
$B^{\ pw}$
$x \overset{R}{\leftarrow} \{1, \ldots, |\mathbb{G}|\}$
$X \leftarrow g^x, X' \leftarrow \mathcal{E}_{pw}(X)$
$A \| X'$
$y \overset{R}{\leftarrow} \{1, \ldots, |\mathbb{G}|\}$
$\mathcal{E}_{pw}(g^y)$
$Y \leftarrow g^y, Y' \leftarrow \mathcal{E}_{pw}(Y)$
$Y \leftarrow \mathcal{D}_{pw}(Y')$
$X \leftarrow \mathcal{D}_{pw}(X')$
$Z \leftarrow Y^x$
$Z \leftarrow X^y$
$sk \leftarrow H(A \| B \| X \| Y \| Z)$

Fig. 3. Encrypted Key Exchange

Whereas the huge majority of the protocols rely on Diffie-Hellman assumptions, some efficient schemes have also been proposed on factoring-related assumptions [36, 37, 25, 30]. Besides the Secure Remote Password (SRP) protocol [39] and the Simple Password Exponential Key Exchange (SPEKE) protocol [32] that have been standardized, EKE-like and GL-like schemes are the two main streams, with security analyses in the UC framework.

3.2 Group Password-Based Authenticated Key Exchange

For groups, while the first proposals were extensions of the group Diffie-Hellman key exchange [38, 20, 17], the Burmester and Desmedt construction [21, 22] became more appropriate, because of its constant number of rounds, independently of the size of the group. Several group password-based authenticated key exchange protocols have then been proposed [2, 9, 1, 5], essentially combining a two-party PAKE with the Burmester and Desmedt methodology.

Acknowledgments. I would like to thank Céline Chevalier for her valuable comments, and all my other co-authors on this quite active and interesting area: Michel Abdalla, Mihir Bellare, Emmanuel Bresson, Dario Catalano, Olivier Chevassut, Pierre-Alain Fouque, Louis Granboulan, Thomas Pornin, Jean-Jacques Quisquater and Phil Rogaway.

References

1. Abdalla, M., Bohli, J.-M., González Vasco, M.I., Steinwandt, R.: (Password) Authenticated Key Establishment: From 2-Party to Group. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 499–514. Springer, Heidelberg (2007)
2. Abdalla, M., Bresson, E., Chevassut, O., Pointcheval, D.: Password-Based Group Key Exchange in a Constant Number of Rounds. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 427–442. Springer, Heidelberg (2006)
3. Abdalla, M., Catalano, D., Chevalier, C., Pointcheval, D.: Efficient Two-Party Password-Based Key Exchange Protocols in the UC Framework. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 335–351. Springer, Heidelberg (2008)
4. Abdalla, M., Catalano, D., Chevalier, C., Pointcheval, D.: Password-Authenticated Group Key Agreement with Adaptive Security and Contributiveness. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 254–271. Springer, Heidelberg (2009)
5. Abdalla, M., Chevalier, C., Granboulan, L., Pointcheval, D.: Contributory Password-Authenticated Group Key Exchange with Join Capability. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 142–160. Springer, Heidelberg (2011)
6. Abdalla, M., Chevalier, C., Pointcheval, D.: Smooth Projective Hashing for Conditionally Extractable Commitments. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 671–689. Springer, Heidelberg (2009)
7. Abdalla, M., Fouque, P.-A., Pointcheval, D.: Password-Based Authenticated Key Exchange in the Three-Party Setting. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 65–84. Springer, Heidelberg (2005)

8. Abdalla, M., Pointcheval, D.: Simple Password-Based Encrypted Key Exchange Protocols. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 191–208. Springer, Heidelberg (2005)
9. Abdalla, M., Pointcheval, D.: A Scalable Password-Based Group Key Exchange Protocol in the Standard Model. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 332–347. Springer, Heidelberg (2006)
10. Barak, B., Canetti, R., Lindell, Y., Pass, R., Rabin, T.: Secure Computation Without Authentication. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 361–377. Springer, Heidelberg (2005)
11. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: 38th Annual Symposium on Foundations of Computer Science, pp. 394–403. IEEE Computer Society Press (October 1997)
12. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)
13. Bellare, M., Rogaway, P.: Entity Authentication and Key Distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994)
14. Bellare, M., Rogaway, P.: Provably secure session key distribution: The three party case. In: 27th Annual ACM Symposium on Theory of Computing, pp. 57–66. ACM Press (May/June 1995)
15. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: 1992 IEEE Symposium on Security and Privacy, pp. 72–84. IEEE Computer Society Press (May 1992)
16. Boyko, V., MacKenzie, P.D., Patel, S.: Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 156–171. Springer, Heidelberg (2000)
17. Bresson, E., Chevassut, O., Pointcheval, D.: Group Diffie-Hellman Key Exchange Secure against Dictionary Attacks. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 497–514. Springer, Heidelberg (2002)
18. Bresson, E., Chevassut, O., Pointcheval, D.: Security proofs for an efficient password-based key exchange. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM CCS 2003: 10th Conference on Computer and Communications Security, pp. 241–250. ACM Press (October 2003)
19. Bresson, E., Chevassut, O., Pointcheval, D.: New Security Results on Encrypted Key Exchange. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 145–158. Springer, Heidelberg (2004)
20. Bresson, E., Chevassut, O., Pointcheval, D., Quisquater, J.-J.: Provably authenticated group Diffie-Hellman key exchange. In: ACM CCS 2001: 8th Conference on Computer and Communications Security, pp. 255–264. ACM Press (November 2001)
21. Burmester, M., Desmedt, Y.: A Secure and Efficient Conference Key Distribution System (Extended Abstract). In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 275–286. Springer, Heidelberg (1995)
22. Burmester, M., Desmedt, Y.: A secure and scalable group key exchange system. Information Processing Letters 94(3), 137–143 (2005)
23. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd Annual Symposium on Foundations of Computer Science, pp. 136–145. IEEE Computer Society Press (October 2001)

24. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.: Universally Composable Password-Based Key Exchange. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer, Heidelberg (2005)
25. Catalano, D., Pointcheval, D., Pornin, T.: **IPAKE**: Isomorphisms for Password-Based Authenticated Key Exchange. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 477–493. Springer, Heidelberg (2004)
26. Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
27. Ding, Y., Horster, P.: Undetectable on-line password guessing attacks. SIGOPS Oper. Syst. Rev. 29, 77–86 (1995)
28. Gennaro, R.: Faster and Shorter Password-Authenticated Key Exchange. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 589–606. Springer, Heidelberg (2008)
29. Gennaro, R., Lindell, Y.: A Framework for Password-based Authenticated Key Exchange. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. Springer, Heidelberg (2003), `http://eprint.iacr.org/2003/032.ps.gz`
30. Gentry, C., Mackenzie, P.D., Ramzan, Z.: Password authenticated key exchange using hidden smooth subgroups. In: Atluri, V., Meadows, C., Juels, A. (eds.) ACM CCS 2005: 12th Conference on Computer and Communications Security, pp. 299–309. ACM Press (November 2005)
31. Groce, A., Katz, J.: A new framework for efficient password-based authenticated key exchange. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 2010: 17th Conference on Computer and Communications Security, pp. 516–525. ACM Press (October 2010)
32. Jablon, D.P.: Strong password-only authenticated key exchange. SIGCOMM Comput. Commun. Rev. 26(5), 5–26 (1996)
33. Katz, J., Ostrovsky, R., Yung, M.: Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 475–494. Springer, Heidelberg (2001)
34. Katz, J., Vaikuntanathan, V.: Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 636–652. Springer, Heidelberg (2009)
35. Katz, J., Vaikuntanathan, V.: Round-Optimal Password-Based Authenticated Key Exchange. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 293–310. Springer, Heidelberg (2011)
36. Lucks, S.: Open Key Exchange: How to Defeat Dictionary Attacks Without Encrypting Public Keys. In: Christianson, B., Lomas, M. (eds.) Security Protocols 1997. LNCS, vol. 1361, pp. 79–90. Springer, Heidelberg (1998)
37. MacKenzie, P., Patel, S., Swaminathan, R.: Password-Authenticated Key Exchange Based on RSA. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 599–613. Springer, Heidelberg (2000)
38. Steiner, M., Tsudik, G., Waidner, M.: Diffie-Hellman key distribution extended to group communication. In: ACM CCS 1996: 3rd Conference on Computer and Communications Security, pp. 31–37. ACM Press (March 1996)
39. Wu, T.D.: The secure remote password protocol. In: ISOC Network and Distributed System Security Symposium – NDSS 1998. The Internet Society (March 1998)