IPHV8I3009X

International Journal Of Advanced Research and Innovation -Vol.8, Issue .III

*ISSN Online: 2319 – 9253*

*Print: 2319 – 9245*

# SECURITY OF THE J-PAKE PASSWORD-AUTHENTICATED KEY EXCHANGE PROTOCOL

[#1]MOHAMMED SALMAN KHAN , M.Tech Student,

[#2]B.SRINIVASA RAO, Assistant Professor,

Dept of CSE,

MOTHER THERESSA COLLEGE OF ENGINEERING & TECHNOLOGY, PEDDAPALLI, T.S.,INDIA.

**ABSTRACT:** J-PAKE is an efficient password-authenticated key exchange protocol that is included in the Open SSL library and is currently being used in practice. We present the first proof of security for this protocol in a well-known and accepted model for authenticated key-exchange that incorporates online and offline password guessing, concurrent sessions, forward secrecy, server compromise, and loss of session keys. This proof relies on the Decision Square Diffie-Hellman assumption, as well as a strong security assumption for the non-interactive zero knowledge (NIZK) proofs in the protocol (specifically, simulation sound extractability). We show that the Schnorr proof-of knowledge protocol, which was recommended for the J-PAKE protocol, satisfies this strong security assumption in a model with algebraic adversaries and random oracles, and extend the full JPAKE proof of security to this model. Finally, we show that by modifying the recommended labels in the Schnorr protocol used in J-PAKE, we can achieve a security proof for J-PAKE with a tighter security reduction.

***Keywords:*** *PAKE protocol, ID-Based Encryption, password.*

## I. INTRODUCTION

In a password-authenticated key exchange (PAKE) protocol, two parties who share only a password (i.e., a short secret) communicate with each other to compute a cryptographically strong shared secret key, using the password for mutual authentication. The protocol should not allow an attacker to obtain any information about the password through simple eavesdropping, and only allow the attacker to gain information about one password per protocol session in an active attack. Basically, this implies that the attacker is not able to obtain data with which to perform an offline dictionary attack, in which the attacker would run through a dictionary of possible passwords offline, checking each one for consistency with the data. A very good introduction and discussion of this problem may be found in Jablon [29] or Wu [47]. The seminal work in the field was the development of Encrypted Key Exchange (EKE) by Bellovin and Merritt [7], [8], and there has been a great deal of work since then (for references see, e.g., [28]).

The J-PAKE protocol [24] is a PAKE protocol that has started seeing wide usage. It is included as an optional protocol in the OpenSSL library [39] (enabled using a config parameter during install, see directory crypto/jpake), and has been used in various products, such as Firefox Sync [16] and Nest products [38] (as part of the Thread protocol [46]). Its popularity is likely due not only to its easy description, straightforward implementation, and practical efficiency, but also because it seems to be based on a different paradigm than previous practical PAKE protocols. Those protocols basically used the password to obfuscate the inputs to a key exchange (e.g., the g x and g y values in a Diffie-Hellman key exchange), whereas the J-PAKE protocol uses ephemeral values like a standard Diffie-Hellman key exchange, but then combines them with a password in an extra round, such that use of the correct password makes certain randomization factors vanish. The JPAKE designers call this the “juggling” technique and attribute the first use of the idea to Hao and Zielinski [25]. Due to its novelty, the designers of J-PAKE claim that it might be useful in avoiding patent issues around other PAKE protocols.

The original J-PAKE paper claimed to give a proof of security, but, as pointed out by Katz [31], the proof was not in one of the well-known accepted models for authenticated key exchange (e.g., the model from Bellare, Pointcheval, and Rogaway [5]), and simply proved some ad-hoc properties in an isolated setting, using implicit assumptions on the adversarial model. Given its growing popularity, it is important to have a better understanding of the security of this protocol, using rigorous and explicit definitions and models. This is especially true for PAKE protocols, since there are many subtleties to their security, and many previous PAKE protocols, or early versions of PAKE protocols (that did not have rigorous security proofs) have been shown to be insecure [36], [41].

In this paper we present a proof of security for the J-PAKE protocol in the well-known authenticated key exchange model of Bellare, Pointcheval, and Rogaway [5], under the Decision Square Diffie-Hellman (DSDH) assumption, along with other assumptions described below. The DSDH assumption is similar to and at least as strong as the Decision Diffie-Hellman assumption, but it is not known whether it is strictly stronger. We note that we could reduce this assumption to DDH and Computational Square Diffie-Hellman (CSDH)1 by using the random-oracle model.2

One interesting technique used in the J-PAKE protocol that has not been used in previous PAKE proofs is the zero- knowledge (ZK) proof of knowledge. Generally it is difficult to argue about the security of ZK proofs of knowledge in a concurrent protocol model. This is because for most known ZK proofs of knowledge, and even non-interactive ZK (NIZK) proofs of knowledge in the random-oracle model, rewinding arguments have been used to prove the extraction property, which is problematic in a concurrent setting since it can cause an exponential expansion in simulation cost during reduction arguments. We initially avoid this issue and assume the use of NIZK proofs of knowledge that are simulationsound extractable [22], with non-rewinding extractors. We call these SE-NIZK proofs. One could say that this proves the security of J-PAKE in a rigorous model that captures the standard intuition behind NIZK proofs of knowledge, and more specifically, proves security under the DSDH assumption and the assumptions necessary to prove the internal NIZK proof of knowledge is simulation-sound extractable.

However, the NIZK proof of knowledge recommended by the designers of J-PAKE (and used in the current implementations) is the Schnorr protocol [43], which seems to require rewinding arguments to prove the extraction property, at least in the standard computation model. Therefore, to provide a rigorous proof of security of J-PAKE using the Schnorr protocol, we turn to the algebraic model [40] (with respect to a group G), in which an adversary is limited to perform only group operations on group elements in G. It is similar to the generic group model of Shoup [44], in which all group operations are performed using an oracle, but is weaker as, in particular, it makes no assumption on the representation of group elements and does not imply by itself that, e.g., the discrete logarithm is hard. We show that in the algebraic model, the Schnorr protocol can be seen as an SE-NIZK proof, in any proof by reduction, with some restrictions (on the group elements used by the proof) that our J-PAKE proof does in fact satisfy. This proof relies on the Discrete Log (DL) assumption in the random-oracle model. Putting this all together, we have proven the security of J-PAKE using Schnorr in the algebraic model and random-oracle model, under the DSDH assumption. It is worth emphasizing that this is a proof of security that matches the underlying implementation in OpenSSL, and this is important in that it allows applications to use J-PAKE in a way that exactly matches the security proof.3

Returning to the standard computation model, Groth, Ostrovsky, and Sahai [23] and Groth [22] show how to achieve SE-NIZK proofs in the common reference string (CRS model). Garay, MacKenzie, and Yang [17] and MacKenzie and Yang [37] show how to achieve non-malleable ZK proofs (which are like SE-NIZK proofs but allowed to be interactive) which trivially imply SE-NIZK proofs in the CRS and random-oracle model, and require only a constant number of exponentiations (but over multiple groups with larger non-prime moduli). Any of these could replace the Schnorr proof of knowledge in the J-PAKE protocol, though none of them would be nearly as practical. As a final result, we show that by slightly modifying the labels used in the Schnorr proofs in the J-PAKE protocol, one can obtain a simpler security proof, with tighter security reductions from known cryptographic assumptions. We recommend using these modified labels in future implementations of the JPAKE protocol, if they don't require backwards compatibility.

***Other PAKE protocols.*** Many previous practical PAKE protocols have been proven secure in either the random-oracle model or ideal-cipher model, e.g., [3], [5], [7], [10], [29], [35], [36]. As shown in [15], [27], the ideal-cipher model is equivalent to the random-oracle model, when the inputs and outputs are binary strings. In practice, however, ideal ciphers for group elements, as required in [5], [7], are difficult to construct and can have an impact on the efficiency of the schemes. In addition, a few PAKE protocols have been proven secure without ideal assumptions. For instance, the practical protocol of Katz, Ostrovsky, and Yung [32] only relies on a reasonably short common reference string that is produced before the protocol begins. This protocol has been generalized and improved in several follow-up works, such as [1], [12], [18], [21], [30], [33]. For these protocols, the common reference string could be simulated using a random oracle. The protocol of Goldreich and Lindell [19] does not rely on a common reference string either, but is only proven secure when protocols sessions are not run concurrently, and does not seem practical. More recently, Goyal, Jain, and Ostrovsky [20] improved the work of Goldreich and Lindell by providing a protocol that is proven secure even when protocols sessions are run concurrently.

## II. RELATED WORK

In Identification process the main focus is improving the security of the authentication system by supplementing it with a secure identification process. To make false login attempts difficult, our method does not use a publicly known login ID for identification. Instead it uses private information known only to the computer system and the user. This process makes the stolen password files unusable for the attackers. In Verification process traditional system have drawback of the password that passwords are stored in a hash table using a cryptographic hash value of the password over a public channel which makes hash value accessible to an attacker. This was Because the password was stored in a single server in hash table, it is not very difficult for a attacker to get the password from a hash value to over increase the strength the two secure channels are necessary for all two-server PAKE protocols, where a password is split into two parts, which are securely distributed to the two servers, during registration. Although we refer to the concept of public key cryptosystem, the encryption key of one server should be unknown to another server and the client needs to remember a password only after registration.

## III. MODEL

For our proofs of security we use a real-or-random variant of the model of [5] with weak adaptive corruptions (corruption queries do not reveal the internal state of the principals, but reveal the password of the principal and can be made at any point during the protocol) and forward secrecy In [2], it is shown that this real-or-random variant is stronger than the original find-then-guess model in [5]. The only difference with [5] is that we allow multiple Test queries.

***Protocol participants and long-lived keys.*** Participants in the protocol are both clients and servers. Each client A holds a password pwA chosen uniformly (and independently) at random from a dictionary of size N. Each server B holds a vector of the passwords of all clients, and when running the protocol with some client A, uses the password pwA of A. Users are modeled as probabilistic poly-time algorithms that respond to queries. For any user U, we will let U denote both the user, and the identifier for the user (e.g., to be used as input to a function).

***Execution of the protocol.*** A protocol P is an algorithm that determines how principals behave in response to inputs from their environment. In the real world, each principal is able to execute P multiple times with different partners, and we model this by allowing unlimited number of instances of each principal. Instance i of principal U is denoted ΠU i . To describe the security of the protocol, we assume there is an adversary A that has complete control over the environment (mainly, the network), and thus provides the inputs to instances of principals. Formally, at the beginning of the protocol, a random bit b is chosen.

## IV. PROPOSED SYSTEM

A system is proposed where security is provided to both the phases of authentication process without the involvement of any specialized devices. The proposed system has two servers. First is Identification Server and second is Verification Server. The proposed system separates the identification server and the verification server, thus it is scalable to a large system. Verification Server is further connected to two more servers named Server 1 and Server 2. To establish a secure channel between verification server and server 1, verification server and server 2 a two way handshaking protocol named Deffie Hellman Key Exchange Protocol is implemented.

In the proposed system the login ID is not considered a secret. The concept of mind metrics is implemented where personal data instead of a login ID to identify a user uniquely. Since it does not accept a login ID during the authentication process, a stolen or cracked password cannot be used for gaining an access to the computing system unless the attacker provides a correct identification material, i.e., mind metrics token. This additional step raises the security of an authentication system. During registration the user submits the token along with login id and other details. The hash value for token is generated and is stored in token database in the tuple format as {token hash value, index}.

All these functions are carried out at Identification Server. In verification process, the concept of two server Password Authenticated Key Exchange (PAKE) protocol, a client splits its password and stores two shares of its password in the two servers, respectively, and the two servers then cooperate to authenticate the client without knowing the password of the client. Each of the password part is encrypted using ElGamal Algorithm and is sent to the respective servers. Both the servers decrypt their part of password information and send it to verification server. The password information is merged at verification server. If merged information and the password entered by the user matches then the particular user gets authentication to the system. We propose a new symmetric solution for twoserver PAKE.

The password pw is secret unless the two servers collude. Although we use the concept of public key crypto system, our protocol follows the password-only model. The encryption and decryption key pairs for the two servers are generated by the client and delivered to the servers through different secure channels during the client registration, as the client in any two- server PAKE protocol sends two halves of the password to the two servers in 4secret, respectively. In fact, a server should not know the encryption key of another server and is restricted to operate on the encryption of the password on the basis of the homomorphic properties of ElGamal encryption scheme. Security analysis has shown that PAKE protocol is secure against passive and active attacks in case that one of the two servers is compromised. Performance analysis has shown that our protocol is more efficient than existing symmetric and asymmetric two-server PAKE protocols.

## V. MULTI-SERVER PAKE PROTOCOLS PAKE

Protocols in the multi-server setting can be classified into two categories as follows. Two-server PAKE: A two-server password-only PAKE protocol was given by Katz [1],which is built upon the two-party PAKE protocol (i.e., the KOY protocol [2]), where two parties, who share a password, exchange messages to establish a common secret key. Their basic two-server protocol is secure against a passive (i.e., honest-but-curious) adversary who has access to one of the servers throughout the protocol execution, but cannot cause this server to deviate from its prescribed behavior. In [23], Katz et al. also showed how to modify their basic protocol so as to achieve security against an active adversary who may cause a corrupted server to deviate arbitrarily from the protocol. The core of their protocol is the KOY protocol. The client looks like running two KOY protocols with two servers in parallel.However, each server must perform a total of roughly 80 exponentiations (i.e., each servers work is increased by a factor of roughly 6 as compared to the basic protocol [2]). Another protocol was given by Yi et al.[3]which propose a new compiler to construct a two-server PAKE protocol with any two-party PAKE protocol.

This compiler employs the two-party PAKE protocol between two servers when they authenticate the client. To achieve the goal,this compiler adds an identity-based encryption (IBE)[4] scheme to protect the messages (containing the password information) from the client to the two servers. The basic idea is: first of all, the client splits its password into two shares and each server keeps one share of the password in addition to a private key related to its identity. In key exchange, the client sends to each server one share of the password encrypted according to the identity of the server. From the client messages, both servers can derive the same one-time password, by which the two servers can run a two-party PAKE protocol to authenticate the client.

This compiler also needs a public key encryption scheme for the servers to protect the messages (containing the password information) from the servers to the client. The one-time public key is generated by the client and sent to the servers along with the password information in the first phase. In an IBE scheme, the decryption key of a server is usually generated by a Private Key Generator (PKG). Therefore the PKG can decrypt any messages encrypted with the identity of the server.Using standard techniques from threshold cryptography, the PKG can be distributed so that the master-key is never available in a single location. In order to prevent a malicious PKG from decrypting the password information encrypted with the identity of a server, a strategy is to employ multiple PKGs which cooperate to generate the decryption key for the server. As long as one of the PKGs is honest to follow the protocol, the decryption key for the server is known only to the server. Since it can assume that the two servers in two-server PAKE never collude, it can also assume that at least one of the PKGs do not collude with other PKGs. Recently, Yi et al. constructed an ID2S PAKE protocol[5] with the Identity-based signature scheme.It propose a new compiler for ID2S PAKE protocol based on any identitybased signature scheme (IBS).

The basic idea is: The client splits its password into two shares and each server keeps one share of the password in addition to a private key related to its identity for signing. In key exchange, each server sends the client its public key for encryption with its identity-based signature on it. The signature can be verified by the client on the basis of the identity of the server. If the signature is genuine, the client submits to the server one share of the password encrypted with the public key of the server. With the decryption keys, both servers can derive the same one-time password, by which the two servers can run a two-party PAKE protocol to authenticate the client. In addition, it generalize the compiler based on IBE in [3] by replacing the Cramer-Shoup public key encryption scheme with any public key encryption scheme. Unlike the compiler based on IBS, the compiler based on IBE assumes that each server has a private key related to its identity for decryption. In key exchange, the client sends to each server one share of the password encrypted according to the identity of the server. In addition, a one-time public key encryption scheme is used to protect the messages (containing the password information) from the servers to the client. The one-time public key is generated by the client and sent to the servers along with the password information in the first phase.In the identity-based cryptography, the decryption key or the signing key of a server is usually generated by a Private Key Generator (PKG).

Therefore the PKG can decrypt any messages encrypted with the identity of the server or sign any document on behalf of the server. Threshold PAKE:Here n severs, sharing the password of the client, cooperate to authenticate the client and establish independent session keys with the client. As long as n - 1 or fewer servers are compromised, their protocol Di Raimondo and Gennaro [6] suggested the first threshold protocols for password authentication which are provably secure in the standard model. This line of research can be thought as applying the tools of threshold cryptography to the problem of password authentication. Threshold cryptography aims at the protection of cryptographic secrets, such as keys, by distributing them across several servers in order to tolerate break-ins.Here the password is shared among a set of n servers so that t of them co-ordinate to authenticate the client.So the adversary can learn the password only by breaking into t+1 of them.It proposes two protocols:Transparent and Non-transparent protocols.In transparent protocol,the client is not aware that at the server's side the protocol has been implemented in a distributed fashion, nor should he know how many servers are involved. The client interacts with a gateway server which to the client's eyes will be the authentication server.

At the end the secret key will be shared between the client and the gateway.To the eyes of the client the protocol should look exactly like a centralized KOY protocol. All the messages exchanged by the client and the gateway will follow the pattern of a regular KOY protocol. In Non-transparent protocol,the client is aware of the distributed implementation of the servers and in particular of the number n of servers.At the end of this protocol the client will establish n separate keys, one with each server. The adversary will only learn the keys established by the corrupted servers.Here the client will basically run n copies of the KOY protocol, one with each server. The servers will cooperate to compute together the answers of the ith server in the ith execution of the KOY protocol.

## VI. CONCLUSION

This is the review of different types of PAKE protocols that are in existence. The comparison based on their performance helps to suit for the respective applications. Adding IBE scheme to the threshold PAKE protocol can be considered as a future work.

## REFERENCES

[1] M. Abdalla, F. Benhamouda, and D. Pointcheval, “Public-key encryption indistinguishable under plaintext-checkable attacks,” in PKC 2015, ser. LNCS, J. Katz, Ed., vol. 9020. Springer, Mar. / Apr. 2015, pp. 332–352. (Pages 2 and 3.)

[2] M. Abdalla, P.-A. Fouque, and D. Pointcheval, “Password-based authenticated key exchange in the three-party setting,” in PKC 2005, ser. LNCS, S. Vaudenay, Ed., vol. 3386. Springer, Jan. 2005, pp. 65–84. (Page 4.)

[3] M. Abdalla and D. Pointcheval, “Simple password-based encrypted key exchange protocols,” in CT-RSA 2005, ser. LNCS, A. Menezes, Ed., vol. 3376. Springer, Feb. 2005, pp. 191–208. (Pages 2 and 3.)

[4] F. Bao, R. H. Deng, and H. Zhu, “Variations of Diffie-Hellman problem,” in ICICS 03, ser. LNCS, S. Qing, D. Gollmann, and J. Zhou, Eds., vol. 2836. Springer, Oct. 2003, pp. 301–312. (Page 5.)

[5] M. Bellare, D. Pointcheval, and P. Rogaway, “Authenticated key exchange secure against dictionary attacks,” in EUROCRYPT 2000, ser. LNCS, B. Preneel, Ed., vol. 1807. Springer, May 2000, pp. 139–155. (Pages 1, 2, 3, and 4.)

[6] M. Bellare and P. Rogaway, “Random oracles are practical: A paradigm for designing efficient protocols,” in ACM CCS 93, V. Ashby, Ed. ACM Press, Nov. 1993, pp. 62–73. (Page 1.)

[7] S. M. Bellovin and M. Merritt, “Encrypted key exchange: Passwordbased protocols secure against dictionary attacks,” in 1992 IEEE Symposium on Security and Privacy. IEEE Computer Society Press, May 1992, pp. 72–84. (Pages 1, 2, and 3.)

[8] ——, “Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise,” in ACM CCS 93, V. Ashby, Ed. ACM Press, Nov. 1993, pp. 244–250. (Page 1.)

[9] D. Boneh, “The decision Diffie-Hellman problem,” in Algorithmic number theory. LNCS, 1998, pp. 48–63. (Page 5.)

[10] V. Boyko, P. D. MacKenzie, and S. Patel, “Provably secure password-authenticated key exchange using Diffie-Hellman,” in EUROCRYPT 2000, ser. LNCS, B. Preneel, Ed., vol. 1807. Springer, May 2000, pp. 156–171. (Pages 2 and 3.)

[11] E. Bresson, O. Chevassut, and D. Pointcheval, “New security results on encrypted key exchange,” in PKC 2004, ser. LNCS, F. Bao, R. Deng, and J. Zhou, Eds., vol. 2947. Springer, Mar. 2004, pp. 145–158. (Page 5.)

[12] R. Canetti, S. Halevi, J. Katz, Y. Lindell, and P. D. MacKenzie, “Universally composable password-based key exchange,” in EUROCRYPT 2005, ser. LNCS, R. Cramer, Ed., vol. 3494. Springer, May 2005, pp. 404– 421. (Page 2.)

[13] C. Chevalier, P.-A. Fouque, D. Pointcheval, and S. Zimmer, “Optimal randomness extraction from a Diffie-Hellman element,” in EUROCRYPT 2009, ser. LNCS, A. Joux, Ed., vol. 5479. Springer, Apr. 2009, pp. 572–589. (Page 4.) 15

[14] Y. Cliff, C. Boyd, and J. M. González Nieto, “How to extract and expand randomness: A summary and explanation of existing results,” in ACNS 09, ser. LNCS, M. Abdalla, D. Pointcheval, P.-A. Fouque, and D. Vergnaud, Eds., vol. 5536. Springer, Jun. 2009, pp. 53–70. (Page 4.)

[15] J.-S. Coron, J. Patarin, and Y. Seurin, “The random oracle model and the ideal cipher model are equivalent,” in CRYPTO 2008, ser. LNCS, D. Wagner, Ed., vol. 5157. Springer, Aug. 2008, pp. 1–20. (Page 2.)

[16] “Firefox Sync.” [Online]. Available: https://www.mozilla.org/en-US/ firefox/sync/ (Pages 1, 5, and 11.)

[17] J. A. Garay, P. D. MacKenzie, and K. Yang, “Strengthening zeroknowledge protocols using signatures,” Journal of Cryptology, vol. 19, no. 2, pp. 169–209, Apr. 2006. (Page 2.)

[18] R. Gennaro and Y. Lindell, “A framework for password-based authenticated key exchange,” ACM Transactions on Information and System Security, vol. 9, no. 2, pp. 181–234, 2006. (Pages 2 and 3.)

[19] O. Goldreich and Y. Lindell, “Session-key generation using human passwords only,” in CRYPTO 2001, ser. LNCS, J. Kilian, Ed., vol. 2139. Springer, Aug. 2001, pp. 408–432. (Page 2.)

[20] V. Goyal, A. Jain, and R. Ostrovsky, “Password-authenticated sessionkey generation on the internet in the

plain model," in CRYPTO 2010, ser. LNCS, T. Rabin, Ed., vol. 6223. Springer, Aug. 2010, pp. 277–294. (Page 2.)

[21] A. Groce and J. Katz, "A new framework for efficient password-based authenticated key exchange," in ACM CCS 10, E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, Eds. ACM Press, Oct. 2010, pp. 516– 525. (Pages 2 and 3.)

[22] J. Groth, "Simulation-sound NIZK proofs for a practical language and constant size group signatures," in ASIACRYPT 2006, ser. LNCS, X. Lai and K. Chen, Eds., vol. 4284. Springer, Dec. 2006, pp. 444–459. (Pages 2, 3, and 11.)

[23] J. Groth, R. Ostrovsky, and A. Sahai, "Perfect non-interactive zero knowledge for NP," in EUROCRYPT 2006, ser. LNCS, S. Vaudenay, Ed., vol. 4004. Springer, May / Jun. 2006, pp. 339–358. (Page 2.)

[24] F. Hao and P. Ryan, "J-pake: Authenticated key exchange without pki," in Transactions on Computational Science XI, ser. Lecture Notes in Computer Science, M. Gavrilova, C. Tan, and E. Moreno, Eds. LNCS, 2010, vol. 6480, pp. 192–206. (Pages 1, 3, 4, and 11.)

[25] F. Hao and P. Zielinski, "A 2-round anonymous veto protocol," in Security Protocols, 14th International Workshop, Cambridge, UK, March 27-29, 2006, Revised Selected Papers, ser. Lecture Notes in Computer Science, B. Christianson, B. Crispo, J. A. Malcolm, and M. Roe, Eds., vol. 5087. LNCS, 2006, pp. 202–211. (Page 1.)

[26] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, "A pseudorandom generator from any one-way function," SIAM Journal on Computing, vol. 28, no. 4, pp. 1364–1396, 1999. (Page 4.)

[27] T. Holenstein, R. Künzler, and S. Tessaro, "The equivalence of the random oracle model and the ideal cipher model, revisited," in 43rd ACM STOC, L. Fortnow and S. P. Vadhan, Eds. ACM Press, Jun. 2011, pp. 89–98. (Page 2.)

[28] D. P. Jablon. [Online]. Available: http://www.jablon.org/passwordlinks. html (Page 1.)

[29] ——, "Strong password-only authenticated key exchange," SIGCOMM Comput. Commun. Rev., vol. 26, no. 5, pp. 5–26, Oct. 1996. (Pages 1, 2, 3, and 6.)

[30] S. Jiang and G. Gong, "Password based key exchange with mutual authentication," in SAC 2004, ser. LNCS, H. Handschuh and A. Hasan, Eds., vol. 3357. Springer, Aug. 2004, pp. 267–279. (Pages 2 and 3.)

[31] J. Katz. [Online]. Available: https://www.lightbluetouchpaper.org/2008/ 05/29/j-pake/#comment-9547 (Page 1.)