Problems with PAKE protocols

Lars Mueller *Technical University Munich*
Munich, Germany
lars.mueller@tum.de

***Abstract*—In the last years data breaches in websites have become fairly common. This happens**

***Index Terms*—**

I. Introduction

notes

- normal password auth Vulnerable to offline Attacks
- motivation for development of pake protocls
- standardization
- Upcomming Questions, why not standard today only a few
- Topics of paper:
- Short Introduction to PAKE Protocols and their cryptography behind them
- The Usage of PAKE in applications today
- Attacks on PAKE
- Some other reasons PAKE isnt used widely

II. Related Work

As PAKE protocls have a long history in somputer science terms, there is a lot of research already being done. There are a lot of different approaches on the topic and different ideas to solve different problems. The first appearance of Encrypted Key Exchange was 1992 in a paper which described a basic protocol secure against dictionary Attacks. The first standardization of PAKE Protocols came with IEEE P1363.2. This project was formed because of huge interest in Industry and Science in the theme During this first so called period of PAKE protocols these protocols where revised and reworked multiple times which extended the working period to 2008. However after the standardization was finsihed it didnt lead to huge adoption in the industry as it was hoped. In the second phase of PAKE development some services adopted the PAKE protocl such as Apple Icloud or Mozzila Firefox. In 2018 WPA3 the replacement for WPA2, the protocol to secure wifi networks, was announced by the WIFI Alliance. It includes an PAKE protocol called Dragonfly to authenticate with the wifi router. The huge amount of PAKE protocls followed the problem

III. Background

A. The basic security principels of PAKE

PAKE basicly allow 2 parties to establish a secure channel inwhich they can communicate without the fear of a 3rd partie to listen. The Requirements are the following

- Resistance against Dictonary Attacks:
  The communitcation between the two parties must not be decryptable. This means that there is not data obtainable which allows an attacker to find the private secret. Especially if the guesses on the secret are run offline by a dictionary or another password-decryption like attack.
- On Password guess per Conection:
  When establishing a new connection between two parties only one guess on the secret is possible. It isnt possible for an attacker to send for example 100 password in one connection attempt, the attacker needs 100 to try them out. Additionally these attempts should be visible and blockable to prevent further guessing the secret.
- forward secrecy:
  Alice and Bob have established a protected session with their pre-shared secret, they both a session-secret which allows them to communicate securly. Eve gets to known the pre-shared secret. She can establish a new session with either Alice or Bob and impersonate the other, however she does not know the session-secret and can therefore not listen to what Bob and Alice are doing in their already established session.
- session-key security:
  Additionally to the session of Alice and Bob there is another session between Alice and Charlie. Eve is now able to obtain the session-secret from Alice-Bob, this means she can listen to their communitcation. The session between Alice and Charlie is still not compromised. This is the case for every other session.

A PAKE is a two stage protocol.

B. Basics of encrypted Communication

1) Safe Prime:

2) Hashing: A hash function takes a key as input. The output is a fixed size hashcode. These functions are used to map data to make it indexable That would be the case if the hashfunction was perfect which is physically not possible. Hashfunctions follow three principles to withstand different types of attacks

- Pre-Image resistance
  It is difficult to find a corresponding message M to a given hash h, $h = hash(M)$. The function is a one-way function.
- Second Pre-Image resistance
  It is difficult to find another message $M2$ getting the same hash as the first message $M1$. $hash(M1) = hash(M2)$.
- Collsion resistance
  Similar to second pre-image resistance, it should be difficult to find two message $M1, M2$ that have the same hash. $hash(M1) = hash(M2)$.

3) Zero-Knowledge-Proof: The Zero-Knowledge-Proof describes a way to proof someone else that you know a secret without ever telling the person the secret. The verifing person knows the secret aswell. The veriefer can ask you different questions which are conducted from the secret, which you can answer correct if you know the secret. This can be repeated until the veriefer is convinced that you know the secret. An Abstract example would be Alice is colorblind and Bob is not. Bob has a red and a green ball. They seem identical to Alice so she is not sure if Bob is telling her the truth and they are different. She wants to proof Bob and holds one Ball in her left the other in her right hand. Bob knows in which hand they are curently. Alice decides, without Bob looking, if she wants to switch the balls or not after shes done that she asks bob if she switched or not. Bob answers, if both balls have the same color bob will eventually choose the wrong option. If they are differently colored Bob shoudl be able to tell Alice if she switchted or not. Like a lot of conecpts in cryptography the zero-knowledge-proof has some properties which define it.

- completeness
  if the proof is correct, the proofer will convice the veriefer that he is correct
- Soundness
  if the proof is wrong, the veriefer will not be conviced by the proofer, however there is a small probabilty for error
- Zero-Knowledge
  There is no secret leeked by proofing.

A famous Zero-Knowledge-Proof would be the Schnoor-Signature

- Group G of prime order q with generator g
- Hash function $H : \{0,1\}^* \leftarrow \mathbb{Z}_q$
- ALICE
- Pick private random key a
- get public key $A = g^a$
- Sign Message $M : \{0,1\}^*$
- 1. Pick Random number r
- 2. Compute $R = g^r$
- 3. Signature $E = H(M, R)$
- 4. Signature $S = r - a \cdot E$
- Send Bob Public Key A, Message M and Signature E, S
- Bob verifys $M : \{0,1\}^*$
- derive $R' = g^S \cdot A^E = g^{r-aE} \cdot (g^a)^E = g^r$
- derive $E' = H(M, R')$
- Check E'=E

C. PAKE Handshake

1) Balanced PAKE: DH-EKE:

- Pre Shared Secret
- A gen. RNR(private key)-¿ publc key -¿ encrypted with PSK
- A send Enc[PSK](public key)
- B dercypt Enc(A) with PSK-¿
- B gen RNR(private key) -¿ publc key
- B gen Sessionkey, random Challenge
- B send Enc[PSK](public key, Enc[Sessionkey](Challenge))
- A decypt, receives bob pub key
- A gen. Sessionkey with her private key and bobs public key
- A decypts 2nd part of message wiht sessionkey
- A generates challenge
- A sends Enc[Sessionkey](challengeA,challengeB)
- B decrypts checks if challengeB is the same (if no sesion is dropped)
- B sends Enc[Sessionkey](challgeneA)
- A decrypts, checks if challgeneA is the same as her
- Can send messages encrypted with sessionkey now

2) Augmented PAKE: SRP:

IV. Usage of PAKE

The newest Wifisecurity standard WPA3 intruducted an Augmented PAKE Protocol to the authentication of secured Wifi networks. This protocl is called Dragonfly and is especially used in the Handshakeprocess of the connection.

A. Problems with PAKE

V. Attacks on PAKE

To explain the attacks and some of their corresponding protocols it is needed to define some variables Let G denotes a subgroup of Z_p^* of prime order q where p is prime and q is big enough for intractability of the Decisional Let $g \in G$ be a generator The hash or the value of the password are in the intervall $[1, p-1]$.

A. Dictonary attack

A pretty basic attack which is pretty common in Public-Key-Infrastructure aswell. To perform a dictionary attack you need a dictionary with common passwords, it is possible to create dictionary based on information from the user. A lot of times dictionary are also generated from leaked passwords on the internet (haveibeenpwnd.com). Next you need an encrypted message or hash which you want to decrypt or find the corresponding message to. The last thing you need is the used encryption/hash function to cipher the message.
Now it is possible to try out as many different passwords as you have in your dictionary, only limited by time and available resources. Even if one security principle of PAKE is that it is protected against offline dictionary attacks some protocls are vulnerable to these attacks. For example SRP, it doenst store the password on the server. it only saves a veriefer, which is a one-way-function of the password hash. This means if the database is breached it would allow an attacker to start an offline dictionary attack on the veriefer. This is pretty similar to todays mostly used Public Key Infrastructure, where only a password hash is stored on the server.

B. pre-computation attacks

Especially the offline dictionary attack is a problem, happens to other services/protocols aswell Assymetric PAKE

C. Impersonation attack

The first EKE Protocls aswell as the SPEKE Protocol is suffering from this attack. Like the name sugests the attacker poses as a user in the impersonation attack to obtain private information. The impersonation attack is applicable if two users have multiple sessions in paralell with each other. Alice and Bob share a common password. Now Alice starts a session with Bob (Session 1) by sending him random seltected number $g^x modp$ (p is a safe prime number). This randomly selected number generated by a genrerator provided a function which takes the shared password as input. After the stage Alice and Bob get their session keys k which is generated from $g^{ab} modp$ For the key verication Alice sends her first key confirmation challenge H(H(k)), where h is a hash function. The prime number g^x and the first key confirmation challenge are intercepted by Eve. Eve raises the prime number by the power of z, a random seltected number. Eve now initatiates another session (Session 2) with Alice using g^{xz}. Alice replies with another random prime number g^y. Eve does the same as before and raises this number by the power of z to g^{yz} and sends it back to Alice. In the next step Eve sends the intercepted key confirmation challenge to Alice which will be answered by Alice with $H(k)$. Now Eve has intercepted Session 1 while owning Session 2 completly which opens up fro different scenarios

D. Replay-Attack

This is a sub-attack of the impersonation attack. The J-PAKE protocol is vulnerable against it. If an Attacker intercepted an earlier run of the protocol it is possible for him to replay these messages to impersonate one party. J-PAKE consists of 4 messages which can be denoted into two independent parts. The Authentication Part is independent from the key exchange part. The second part of J-PAKE which allows to run a replay-attack on it is that the function to generate a session-key doesnt include some session-specific information. This allows for some

E. Time-Attack

F. Invalid-curve attack

VI. Evaluation & Discussion

- a lot of protocols have flaws even the used ones (WLAN) - Normal password based protocols are easier to implement and have the same security for the user, if done right - patents have made it difficult to use PAKE protocols as they are all protected - Pseudo Randomness made it insecure in webbbrowsers

VII. Conclusion

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[8] http://ijns.jalaxy.com.tw/contents/ijns-v17-n5/ijns-2015-v17-n5-p629-636.pdf

[9] Mathy Vanhoef and Eyal Ronen, “Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd,”

[10] https://chunminchang.gitbooks.io/j-pake-over-tls/content/pake/balanced/dh-eke.html

[11] https://www.dcs.warwick.ac.uk/ fenghao/files/pw.pdf

[12] https://eprint.iacr.org/2014/585.pdf