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 protocols
- 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 protocols 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 protocol 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 protocols 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) 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. $hash(M1) = hash(M2)$.

2) 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: SRP is probably the most used protocl of the PAKE protocols. It is used by the mail provider ProtonMail and in the smart home solution of Apple, Apple HomeKit. Another protocol which will probably used in the future a lot is the Dragonfly protocl used in the new WIFI security standard WPA3.

D. Problems with PAKE

IV. 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 protocols are vulnerable to these attacks. An augmented PAKE liek SRP, is not storing the password on the server. Only a veriefer is saved, 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. Nonetheless this does not mean that the SRP protocol is completly secure against dictonary attacks, it is only secure against server data leaks. For example

B. pre-computation attacks

Pre-Communication attacks are a form of dictonary attack, which is like the dictionary attack not limited to PAKE protocols. It works by pre-computing password hashes with

a salt. After a data breach on the server-side or another way that leaks the password hash it is possible to look up the corresponding password instantly. Even though an augmented PAKE isnt directly vulnerable to such attacksm, it is sometimes still possible. An Example would be the Secure Remote Password (SRP) protocol. It is a rather famous protocol because it is used by some applications, therefore there are a lot of different implementations for it. This protocol does not store the password on the server only a veriefer to verify the password. But it is possilbe to pre-compute veriefer for passwords if it is known that the target server is using the SRP protocol. Lets continue with the example of the dictionary attack

C. side-channel attack

This is a group of attacks, which try to find secrets by gathering information from the target. They do not aim on the design of the protocol, they target the implementation These information can be of physical nature like power consumption, electro magnetic radiation. Measuring the time it needs to compute a cryptographic function and than relaying information on the secret is also a sidechannel attack. Aswell as watching the access of the chache used by some victim.

D. PARASITE

While we explained some theory based attacks now we will focus on an attack on the OpenSSL implementation of SRP. This attack could lead to a password leak and is called PARASITE (PAssword Recovery Attack against Srp implementations in ThE wild) and is a cache based side channel attack. In this attack the target is the password of a victim which is saved on the memory. Especially the Flush+Reload and Performance Degradation Attack (PDA) is used. It monitors the memory adresses and finds the used ones by the victim. This is achieved by flushing and reloading the cache and measuring the time needed for the victim to receive the data. If the memory is flushed at the point of access the data needs to be looked up from the memory directly which subsequentlly takes longer. In the OpenSSL implementation of the modular exponentiation, they use a special variable to speed up the square and multiplication process. However the variable owerflows,after that a new bigger variable with the Montgomery representation is used. The Montgomery representation is an eliptic curve used to speed up the process of modular exponentiation. Hence this part oft the implementation allows an attacker to distinguish every iteration on it, as the first part of the implementation was to fast. This makes the the amount of iterations guessable and you therefore guess some bit patterns. The chosen generator is 5, because it is the most convienient one to used. With knowing the generator we can identify on which bits the accumulated variable is overflowing and how many iterations are needed until the overlow occours. Through FLUSH+RELOAD attacks it is possible to deduct some bits of the password while other remain unkown however due to known possition of said bits it is possible tu reduce the remaining possiblities. If it is not possible to find the password with one run of the Protocol, it can be run more times, with another salt, to get different bit patterns of the same password. Because we know some parts of the password it is possible to start a dictionary attack on the password even a pre-computation attack as we know the used salt. With knowing the password we can impersonate both parties. We know the password to tell the server we are the user he expects aswell as telling the user we are the server because we could calculate the verifier stored on the server. These type of attacks happen and are partly possible on differnt cryptographic protocols. Also it is important to mention that the threat model of the attack. It needs access to the memory and the cache of the processor. This is possible through a spy programm on the pc or a java-script injection in the browser The OpenSSL libary is one of the most used crypto libaries and therefore used as reference in other implementations. This is also the case on other implementations of the srp protocol which made them vulnerable aswell. To make the attack a bit more clear

E. Impersonation attack

The first EKE protocols aswell as the SPEKE protocol is suffering from this attack. EKE and SPEKE are balanced PAKEs. 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

F. Replay-Attack

This is an attack on the J-PAKE protocol, which was used in the mozzila firefox syncronisation feature. It is related to the impersonation attack, becuase it replays eavesdropped messages and impersonate the other party. This attack is limited by the fact that the session key cannot be computed from the eavesdropped messages without the password of a session member. it means, that attackers are only authenticated but they cannot encrypt or decrypt messages from the other participant. Basically this is a varaiation of the impersonation attack on augmented PAKEs 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. We assume an Attacker Eve intercepted an earlier run of the protocol and he wants to impersonate Alice. E sends a derived message to B. B proceeds to start his part of the J-PAKE by seleting random numbers and computing Zero-Knowledge-Proofs to them. E answers with the next eavesdropped and derived message. B now verifys the numbers of E with his own and generates a session key k, using values from the first message and his own random generations. Bob thinks Eve is Alice but Eve cannot compute the session key because she doesnt have the password of Alice and x_2. x_2 is the second random generated value of Alice which is never sent to Bob because it was only used as exponent of a save prime number.

G. Dragonblood

Is a series of Vulnerabilieties on the balanced PAKE Dragonfly. This PAKE is used in WPA3 as Handshake.

V. Evaluation & Discussion

Despite their long history PAKE protocols have not jet made the jump to be a well known solution to the common problem of authenticating a user to a server. The current standard is involving a TLS connection which prevents man-in-the-middle attacks, aswell as saving the password of the user on the server. In the best case password is saved on the server as a salted hash. Worst case would be that it is saved or even logged somewhere in clear text. This flaw can also be found in a lot of balanced PAKEs, because they share the same password with each other. With Balanced PAKEs having the problem to rely on saved passwords on server side, augmented PAKEs entered the display. Another important issue with PAKE not used widely, is that most PAKE protocls used to have a patent and that stopped them from being used at all. Nowadays most patents are expired. J-PAKE was the first protocol that was publicly available and theefore used in some applications. Especially Mozilla tried to integrate it into their browsers sync feature, but it got removed shortly after. The same happend with an OpenSSL implementation of J-PAKE it got removed. J-PAKE was a balanced PAKE, it did not provide any advantage over the established methods. These advantages are While the PAKE protocols where under the protection of patents other solutions where developed. The most famous and used other solutions are SSL and their, now standard, successor TLS. Implmenting these protocols in todays services would not change much for the user as he is still required to choose a password. One important factor however is the strength of the used password as the most actual versions (OPAQUE) of PAKE protocols provide enough protection from bruteforce attacks in case of a server breach. This is a good thing as users tend to use simple passwords for their logins which provides a huge security risk for them. So implemting these protocls would not change anything for the user but it will help strengthen the security of the user of a potential password leak. With implemting these protocls the next problem occours: There could be attacks found on the implemtation, like the PARASITE attack on the SRP protocol. On the other hand todays standard protocols are not immunne to attacks either, like the Heatbleed attack on the TLS protocol showed in 2014. Implemting a PAKE in your login flow has the side effect of being resource intensive on the user device. It is easier to compute a salted hash of a password on the device than to compute serveral hashes combined with multiple additional exponential modular calculations. Some PAKE protocls already found their way in the most used crypto libary called OpenSSL. When optimized it should be possible that todays used devices are able to handle the additional workload provided by using a PAKE protocol. These protocols make a pretty strong case for them nowadays especially combined with the already established TLS protocol they are a good and more secure alternative to established methods of authenticatin a user. That PAKE protocols can be used in wireless networks is also showed in by Apple by using the SRP Protocol in their smart home solution.

VI. Conclusion

With WPA3 their is a PAKE underway to come to every wifi network, while it still has some flaws it is nowadays a better solution than WPA2. With a new generation of PAKE there is a possiblity that more serivces and devices will adapt these protocols and be therefore more secure. The past showed that there are a lot of possible threats on PAKE protocols. These threats an vulnerabilieties can be exploited as it was shown by the PARASITE and Dragonblood attacks. Our contribution was to get a short introdutcion to PAKE protocols and show their weakness and strengths. Aswell as providing some history on why PAKE protocols are not adapted by industry despite being a superior solution. We showed that PAKE protocols are not immune to attacks they say they should be, like the SPEKE or EKE protocol against dictionary attacks. As implementations of PAKE protocls becoming standardized in other protocols it is important for them to be secure. It was important for them to be performant which opened them up for side-channel attacks. While still important with todays availablity of computing power it should be possible to implement PAKEs performant and secure. As a final thought, while still not standard PAKEs are on a good way to become more famous as the topic gets more traction. We wanted to contribute to that possible traction by giving the people some insight on why PAKEs are not widely used. Being a superior idea but still having some child troubbles PAKEs still have a long way to go until they are a standard.

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
[13] https://blog.cryptographyengineering.com/2018/10/19/lets-talk-about-pake/
[14] https://eprint.iacr.org/2021/553.pdf
[15] https://eprint.iacr.org/2018/163.pdf
[16] https://eprint.iacr.org/2012/021.pdf