The Security Evaluated Standardized Password Authenticated Key Exchange (SESPAKE) Protocol

draft-smyshlyaev-sespake-09

Stanislav V. Smyshlyaev, Ph.D.
Head of Information Security Department, CryptoPro LLC

Original motivation for SESPAKE development

Token/smart card usage in hostile environments

- Bluetooth tokens, NFC smart cards.
- Remote usage of cryptographic tokens.

Authenticated secure channels without certificates

- User access to digital signature servers.
- Messengers.

General PAKE requirements

- Impossibility for an active adversary to obtain criteria for password
- Implementations protected against side-channel attacks etc.
- ... and everything else defined in J.-M. Schmidt, «Requirements for PAKE schemes», draft-irtf-cfrg-pake-reqs-05.

Original motivation for SESPAKE development

Additional requirements

- Strong performance requirements (due to the need of token usage).
- Explicit key authentication (key confirmation).
- Clear requirements for implementations properties that are crucial for the security.
- Security proofs:
  - complete;
  - open-access;
  - security based on fundamental problems;
  - security level comparable to: 256-bit ECDSA/EC-RDSA;
  - practice-oriented;
  - suitable for security evaluation of end products.

These features are adorable for any PAKE applications, aren't they?

TC 26 document: the SESPAKE protocol

Technical Committee for standardization "Cryptographic mechanisms for information protection"of the Russian standardization system (TC 26).

"Standardization recommendations. Password-based authenticated key establishment protocol."

Public Information: $I, P, Q_1, \ldots, Q_l, m, q$		
$A\ [A_{ID}, PW]$		$B\ [B_{ID}, Q_{PW}, ind, salt]$
	$\xrightarrow{A_{ID}}$	
	$\xleftarrow{B_{ID}, ind, salt}$	
$z_A = 0$		
$Q^A_{PW} = F(PW, salt, 2000) \cdot Q_{ind}$		
$\alpha \in_R \{1, \ldots, q-1\}$		
$u_1 = \alpha \cdot P - Q^A_{PW}$	$\xrightarrow{u_1}$	
		if $u_1 \notin E \Rightarrow$ FINISH
		$z_B = 0$
		$Q_B = u_1 + Q_{PW}$
		$\beta \in_R \{1, \ldots, q-1\}$
		if $\frac{m}{q} Q_B = 0_E \Rightarrow Q_B = \beta P,\ z_B = 1$
		$K_B = H_{256}((\frac{m}{q} \cdot \beta \mod q) Q_B)$
	$\xleftarrow{u_2}$	$u_2 = \beta \cdot P + Q_{PW}$
if $u_2 \notin E \Rightarrow$ FINISH		
$Q_A = u_2 - Q^A_{PW}$		
if $\frac{m}{q} Q_A = 0_E \Rightarrow Q_A = \alpha P,\ z_A = 1$		
$K_A = H_{256}((\frac{m}{q} \cdot \alpha \mod q) Q_A)$		
$tag_A = T_A \Vert A_{ID} \Vert ind \Vert salt \Vert u_1 \Vert u_2$ $M_A = \mathrm{HMAC}_{K_A}(tag_A)$	$\xrightarrow{M_A}$	
		$tag = T_A \Vert A_{ID} \Vert ind \Vert salt \Vert u_1 \Vert u_2$ $M = \mathrm{HMAC}_{K_B}(tag)$
		If $M \neq M_A$ or $z_B \neq 0 \Rightarrow$ FINISH
		$tag_B = T_B \Vert B_{ID} \Vert ind \Vert salt \Vert u_1 \Vert u_2$
	$\xleftarrow{M_B}$	$M_B = \mathrm{HMAC}_{K_B}(tag_B)$
$tag = T_B \Vert B_{ID} \Vert ind \Vert salt \Vert u_1 \Vert u_2$		
$M = \mathrm{HMAC}_{K_A}(tag)$		
If $M \neq M_B$ or $z_A \neq 0 \Rightarrow$ FINISH		

Key generation

Authentication

1 ECDHE: password-protected under S-QPCCDH

2 Established key

3 Reflection resistant key confirmation

«R1: A PAKE scheme MUST clearly state its features regarding balanced/augmented versions.»

- SESPAKE, SPAKE2, DragonFly: balanced.
- AugPAKE, SPAKE2+: augmented.

Augmentation = additional tasks for an adversary in two attack scenarios:
The attacker gets the stored user password-related information from the server, doesn't spend his resources to brute-force PW from f(PW) and impersonates the user on ...

1. ... other servers, where the user used the same PW.
2. ... the same compromised server.

SESPAKE addresses the first attack scenario and does not address the second one.

«R2: A PAKE scheme SHOULD come with a security proof and clearly state its assumptions and models.»

- SESPAKE:
  - proof for session key security: YES
  - proof for key confirmation status: YES
- AugPAKE:
  - proof for session key security: YES (???)
  - proof for key confirmation status: CURRENTLY NO
- SPAKE2:
  - proof for session key security: YES
  - proof for key confirmation status: NO (step undefined)
- DragonFly:
  - proof for session key security: YES
  - proof for key confirmation status: NO

«The proof must show that the probability of an active adversary to pass authentication, to learn anything about the password or to learn anything about the established key equals, up to a negligible term, the chance of randomly guessing the password, while each guess requires an interaction with a legitimate party.»

The security is proven in the indistinguishability-based model for the two threats: obtaining some information about the session key and false-positive key confirmation.

Do we need the key confirmation step in the document?

If we leave the key confirmation step definition „to be added when needed“, we can obtain (at least) a situation similar to SPEKE: a lot of subtle attack scenarios that can lead to real problems with the end solution.

Do we need a separate security proof for it?

If we leave a key confirmation procedure without the security proof, we'll lack the confidence that there won't be the following situation:

- Key agreement is secure: an adversary cannot get any information about session keys or criteria for offline password dictionary attack.
- Key confirmation is weak: an adversary can make the server believe that he posesses the session key.
- Then the server wouldn't detect attack and wouldn't increment his false authentication attempts counter — letting the adversary to continue his online password guessing attacks.

«R2: A PAKE scheme SHOULD come with a security proof and clearly state its assumptions and models.»

$$\mathrm{Adv}_{\mathrm{SESPAKE}}(t, q_{\mathrm{exec}}, q_{\mathrm{send}}, q_{\mathrm{reveal}}, q_H) \leqslant$$

$$\leqslant \frac{4}{|D|} q_{\mathrm{send}}^2 + 2q_{\mathrm{send}} \frac{(2q_{\mathrm{exec}} + q_{\mathrm{send}})^2}{q} + 4q_{\mathrm{send}} q_H \mathrm{Adv}_{\mathrm{CDH}}(t + 2\tau q_{\mathrm{exec}}) +$$

$$+ 4q_{\mathrm{send}}^2 \sqrt[6]{\frac{\mathrm{Adv}_{\mathrm{SCDH}}(4t + \Theta + O(q_H \tau))}{|D|^2} + \frac{2^{13} q_H^4}{|D|^2 q}}.$$

«R3: The authors SHOULD show how to protect an implementation of their PAKE scheme in hostile environments, particularly, how to implement their scheme in constant time to prevent timing attacks»

The recommendations are given — including the most important part of ones that are needed for certain special cases of temporary points becoming (due to a kind of very specific MitM attack and a curve of composite order) zero points — in these cases a fake calculation scenario that would prevent timing attacks is described (although the correct way of handling the counters would prevent the attacks).

«R4: In case the PAKE scheme is intended to be used with ECC, the authors SHOULD discuss their requirements for a potential mapping or define a mapping to be used with the scheme.»

The recommendations are given.

«R5: A PAKE scheme MAY discuss its design choice with regard to performance, i.e., its optimization goals.»

If the server has limited computing resources it can store a point Q_{PW} instead of the password PW in order to skip the step of the time-consuming computation of the point Q_{PW}. There are recommendations for the optimized version of the SESPAKE protocol that have been given during the CFRG discussion and can be added to the draft itself.

«R6: The authors of a scheme MAY discuss variations of their scheme that allow the use in special application scenarios. In particular, techniques that allow agreeing on a long-term (public) key are encouraged.»

The protocol can be used for secure channel establishment — the core issue here is the allowed security level. These questions are discussed in the paper with security proofs and can be added to the draft itself.

«R7: A scheme MAY discuss special ideas and solutions on privacy protection of its users.»

The property is discussed in the CFRG mailing list and can be added to the draft itself. It can be achieved with some additional measures: the usage of the PKC or the storage of the additional long-term values. Mechanisms can be implemented with the protocols that are independent of the SESPAKE scheme.

«R8: The authors MUST declare the status of their scheme with respect to patents.»

This protocol is approved in the standardization system of the Russian Federation, has no patent and is available for free use.

The procedures of handling the counters are crucial for the security of the implementations.

- If the false authentications counter is not handled before the start of the protocol, any exceptions that are handled in an inappropriate way would lead to practical exploits.
- Absence of a counter of false authentication attempts in a row can make resistance to DDoS attacks impossible.
- Misuse of counters of total false authentications can open a way to online attacks of the following type: m authentication attempts by adversary followed by 1 legal authentication — in a loop.

- The SESPAKE draft contains all details related to the handling of the counters, incorrect parameters etc.
- The security estimations for implementations satisfying the document follow immediately from the security proof — without rough edges between the security of “paper“ protocol and real implementations.

Performance

PAKEs allow to precompute several items to reduce online phase for the following interactions.

For optimized versions of protocols the following running times (if independent operations run in parallel) can be obtained (note: these values are about the total time, not the complexity):

	AugPAKE	SESPAKE	SPAKE2	DragonFly
KDF	0	0	–	1
Elliptic curve points scalar multiplications	3	1	1	2
Hash compression function (256-bit hash and curve)	11	14	–	10

Conclusion

SESPAKE core features

- All requirements of draft-irtf-cfrg-pake-reqs-05 are met.
- The security is proven for the full protocol: both for key agreement and key confirmation parts.
- The security proofs contain practice-oriented details: the security estimations for the particular implementations follow.
- The issues that are crucial for the security of implementations are addressed in the document.
- Reduced total running time of the online phase.

draft-smyshlyaev-sespake-09

The Security Evaluated Standardized Password Authenticated Key Exchange (SESPAKE) Protocol.

Remaining questions

- Should we include the optimization techniques in the document?
- Should we describe common anonymization methods in the document?
- Should we include comments about security against specific subtle attack scenarios (cf. attacks on SPEKE)?
- Should we define some „default“ algorithms and parameters (e.g. SHA-3 and Ed25519 curve) for use with the protocol?
- The current examples now are for the Russian Stribog hash and elliptic curves — for which algorithms should we include examples in the final version?

Thank you for your attention!

Questions?

- Materials, questions, comments:
  - svs@cryptopro.ru