Responses to PAKE Selection – Additional Information about VTBPEKE (Revised)

Guilin Wang (wang.guilin@huawei.com) and David Pointcheval (david@pointcheval.fr)

August 16, 2019

As required by CFRG, we are here to give the reviewers (on the future steps of the process) as much information about VTBPEKE, as possible.

Part A: How Are the Eight PAKE Requirements Meet?

"a. expanded answers for all positions of RFC 8125 regarding their PAKEs"

Responses: According to RFC 8125, we are here to answer how VTBPEKE meet the eight requirements as follows, in our best.

REQ1: VTBPEKE is an augmented PAKE scheme. It has a balanced version as well, called TBPEKE, which is more efficient but not resistant to server compromise attack.

REQ2: VTBPEKE does come with a security proof, in the Real-or-Random security model under the complexity assumptions that GDH (Gap Diffie-Hellman) and GSDH (Gap Simultaneous Diffie-Hellman) problems are hard. More details can be found from [PW17].

REQ3: VTPEKE may follow the usual ways to protect its implementation in hostile environments. In particular, the scheme is constant time up to constant-time implementation of hash function and exponentiation evaluations.

REQ4: VTBEKE is intended to be used with ECC due to efficiency consideration, and it just needs normal hash functions, not a mapping.

REQ5: The main motivation of VTBPEKE is to achieve provable security for PAKE when ECC is used, without efficiency lost compared to existing schemes, but does not target to particular performance optimization goals. We will consider this to see if the scheme can do better.

REQ6: We will consider possible variations of VTBPEKE, according to the suggestion given in RFC 8125.

REQ7: In normal cases, a public key can be used to encrypt a user's identity. We are happy to consider if there are other solutions on privacy protection of its users.

REQ8: We are happy to follow the IRTF IPR policy <https://irtf.org/ipr>.

Part B: Further Questions

"b. their own opinions on the following questions (they does not need to be complete: for example, a designer of a PAKE might not be an expert in TLS and might not be able to reply how his PAKE can be incorporated in TLS 1.3):"

- "How does it meet the "SHOULD" requirements of RFC 8125?"
- **Response**: VTBPEKE meets two "MUST" requirements of RFC 8125*, i.e.*, REQ1 and REQ8 and the three "SHOULD" requirements of RFC 8125, *i.e.*, REQ2, REQ3, and REQ4.

- "Does it meet "crypto agility" requirements, not fixing any particular primitives and/or parameters?"
- **Response**: Yes, VTBPEKE meets "crypto agility" requirements, as it does not fix particular primitives and/or parameters.

- "What setting is the PAKE suitable for, which applications does it have? "
    - "Peer communication" (where both ends share the same password) or "client-server" (where the server does not store the password but only a one-way function of the password)?
    - **Response**: VTBPEKE is an augmented PAKE scheme, so it targets for "client-server" applications. However, it has a balanced version as well, called TBPEKE, which can support "Peer communication" applications. In fact, TBPEKE is more efficient but not resistant to server compromise attack. Refer to [PW17] for detail.

    - "A nomination should specify for which use-cases the protocol is recommended ("PAKE as a more-secure replacement for a PSK on a machine-2-machine interface" or "PAKE for securely accessing a remote HMI interface server (e.g. a web server) without configured WEB-PKI certificates")."
    - **Response**: VTBPEKE is mainly a "PAKE as a more-secure replacement for a PSK on a machine-2-machine interface". For the latter case, we will happy to check if it can be supported as well.

    - "Can two communicating parties initiate the key exchange process at the same time, or must it always be the case that only one party can initiate the process?"
    - **Response**: In VTBPEKE, only the client side initiates the key exchange process. But the balanced TBPEKE version could support simultaneous flows from the two parties.

    - "Is it suitable to be considered as a standalone scheme?"
    - **Response**: Yes, VTBPEKE is suitable to be considered as a standalone scheme.

    - "Can it be integrated into IKEv2? TLS Handshake? Any other protocols?"
    - **Response**: Yes, we believe so.

- "Is there a publicly available security proof? If yes,
    - Are there known problems with the proof?
    - Is the considered security model relevant for all applications that PAKE is intended for (e.g., if a PAKE is to be used in TLS Handshake, it is important that the TLS adversary model is considered for the PAKE)?
    - Does it allow to be sure in sufficient level of security for common values of password lengths?"
    - **Response**: Yes, VTBPEKE has a publicly available security proof [PW17]. The Real-or-Random security model used for VTBPEKE is comparable to the TLS adversary model, as we know. A good feature of VTBPEKE is that multiple parameters are recommended for different practical security levels with different security requirements (e.g., forward security is required or not). So it does allow to be sure in sufficient level of security for common values of password lengths.

- Security assessment.
    - "Does its security depend on any nontrivial implementation properties? Are they clearly stated in the document? "
    - **Response**: No, VTBPEKE does not depend on any nontrivial implementation properties.

    - "Does the PAKE have precomputation security (for augmented PAKEs)?"
    - **Response**: Not very sure what precomputation security is, but the encoded password on the server-side is just an exponentiation of the hash of the password with a salt, which is transmitted by the client.

    - "If the PAKE relies on the assumption of a trusted setup - more specifically, the discrete logarithm relationship between the two group elements in the system setup MUST be unknown - in anticipation of the worst but not impossible scenario, the authors should clearly state the security implications when the discrete logarithm relationship becomes known, and the subsequent mitigation measures."
    - **Response**: TBPEKE and VTBPEKE require two independent group elements that can be generated once for all from a hash function, until one falls in the group (which is efficient on Elliptic Curves). If their relative discrete logarithm would become known, all the security vanishes, as a dictionary attacks becomes available.

- Performance assessment.
    - "What's with the "round efficiency" of the PAKE? In a standard two/multi-party secure computation setting, the "round" is defined as a step in which all parties can complete operations at the same time without depending on each other. In practice, a 2-round protocol could be implemented as 2 flows or 3 flows depending on the application context, but that's more the implementation detail."
    - **Response**: VTBPEKE is a 3-round protocol.

- "How many operations of each type (scalar multiplications, inversions in finite fields, hash calculations etc.) are made by each side?"
- **Response**: For running one time of VTBPEKE, the following operations are needed to perform:

Client Side			
Hashing	Multiplications / Scalar multiplications	Exponentiations /Point Multiplication	Symmetric Key Encryption
2	1	4	1
Server Side			
Hashing	Multiplications / Scalar multiplications	Exponentiations / Point Multiplications	Symmetric Key decryption
1	0	4	1

- "Which recommendations for secure usage can be given?"
  - "Is it defined how the explicit key confirmation is performed/must be performed externally? Are there clear statements whether this procedure is optional or mandatory?"
  - **Response**: By default, VTBPEKE is with explicit key confirmation for the server. The client should perform an additional check.
  - "Can any recommendations on using iterated hashing (e.g., with Scrypt) with the PAKE be given?"
  - **Response**: In VTBPEKE, the hashing can be any standard one, including iterated hashing, like Scrypt.
  - "Can any recommendations to avoid a user enumeration attack be given?"
  - **Response**: Not very sure what user enumeration attack is, but it is proven to resist to dictionary attacks for outsider adversaries (possibly active adversaries). Of course, the server can do an exhaustive search. But an iterated hashing (Scrypt or any PBKDE) can improve security with respect to the server, or in case of server-compromise.

Part C: References

[PW07] David Pointcheval and Guilin Wang. VTBPEKE: Verifier-based Two-Basis Password Exponential Key Exchange. In the Proc of AsiaCCS 2017, pp. 301-312, ACM Press, 2017. Author Version is available at: https://www.di.ens.fr/david.pointcheval/Documents/Papers/2017_asiaccsB.pdf