Flaws of a Password-Based Three-Party Authenticated Key Agreement Protocol in Post-quantum Environment

Sonam Yadav, Vivek Dabra, Pradeep Malik, and Saru Kumari

1 Introduction

Recently, Islam and Basu [3] proposed a password-based three-party authenticated key agreement protocol for mobile devices in a post-quantum environment (PB-3PAKA) protocol. The formal security of the PB-3PAKA protocol is demonstrably secure in Random Oracle Model (ROM). The PB-3PAKA [3] protocol establishes a session key between two mobile users using fresh pair of keys in every session.

Key computation and communication costs are costly, so the key reuse is known to enhance performance during real-world deployments to cut the cost. The resumption mode in TLS v1.2 permits key reuse which drastically decreases online computations. An efficient 0-round-trip time (RTT) resumption mode is suggested in TLS v1.3 draught version 7 [5]. It allows TLS to establish a secure connection without incurring round-trip costs. According to TLS v1.3 version 7, the majority of key exchange computations and communication costs are saved by reusing public and private key pairs. Resumption mode is used to establish the overwhelming majority of TLS connections in the real world. But, this feature-induced security vulnerability in the existing post-quantum key exchange protocol. The key reuse vulnerability has been first identified by Kirkwood et al. [4] in the post-quantum environment. In their work, the reuse of public/private keys is shown to break the security of the protocol. Therefore, authors [4] advise that public-key validation is necessary for the RLWE-based key agreement protocol.

Ding et al. [1] proposed an attack. This attack is known as a signal leakage attack (SLA) and it is against the RLWE-based reconciliation schemes where the public/

S. Yadav (✉) · P. Malik
Department of Mathematics, Faculty of Science, Shree Guru Gobind Singh Tricentenary University, Gurugram 122505, Haryana, India
e-mail: sonamyadav20jan@gmail.com

V. Dabra
Department of Computer Science and Engineering, Panipat Institute of Engineering and Technology, Panipat 132103, Haryana, India

S. Kumari
Department of Mathematics, Chaudhary Charan Singh University, Meerut 250004, Uttar Pradesh, India

B. K. Roy et al. (eds.), *Cryptology and Network Security with Machine Learning*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-99-2229-1_6

private keys are reused. In this approach, the adversary initiates multiple sessions with the honest party to recover the honest party's private key. Using a $2.q$ number of queries with the honest party, the adversary can recover the honest party's secret key.

Continuing the above work, Ding et al. [2] improved signal leakage attack (SLA), and this improved attack is known as i-SLA, in which $2.q$ number of queries is reduced to q number of queries. Now, the secret of the reused public key of the honest party can be recovered with fewer queries.

Influenced by these researchers, we found that Islam and Basu's [3] proposed protocol is vulnerable to dishonest user's attack, signal leakage attack.

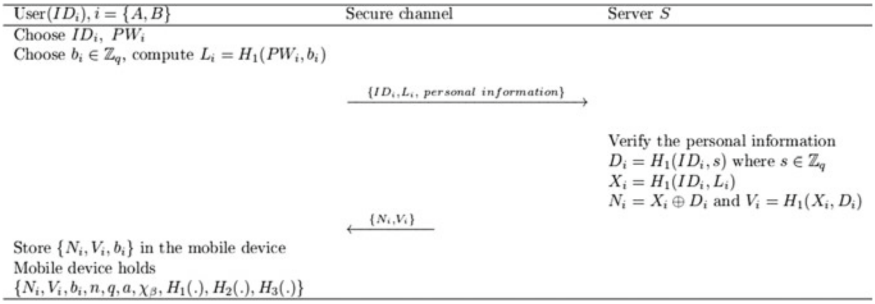
2 Review of Islam and Basu's Protocol

In this section, we introduce Islam and Basu's PB-3PAKA protocol [3]. The PB-3PAKA protocol has four phases: initialization phase, user registration phase, authenticated key agreement phase, and password change phase.

Table 1 shows the notations of Islam and Basu's PB-3PAKA protocol. Figures 1 and 2 describe the user registration and authenticated key agreement phase, respectively.

Table 1 Notations of Islam and Basu's [3] protocol

Notation	Meaning
q	Large prime number
a	Random element sampled from R_q
χ_β	Discrete Gaussian Distribution
A/B	Initiator/Responder
S	Server
$\mathcal{A}$	Adversary
x_i	Public key of i, $i \in \{A, B\}$
r_i	Secret key of i
s	Server's secret key
Cha	Characteristic function
Mod_2	Modular function
ID_i	Identity of U_i/S
PW_i	Password of User i
SK	Session key
$\mathbb{Z}$	Set of integer numbers
$\mathbb{Z}_q$	$\mathbb{Z}$ modulo q
$\oplus$	Bitwise XOR
$H(.)$	Collision resistance function
D	Password dictionary, where $PW_i \in D$



User$(ID_i), i = \{A, B\}$ | Secure channel | Server S

Choose ID_i, PW_i

Choose $b_i \in \mathbb{Z}_q$, compute $L_i = H_1(PW_i, b_i)$

$\xrightarrow{\{ID_i, L_i,\ personal\ information\}}$

Verify the personal information

$D_i = H_1(ID_i, s)$ where $s \in \mathbb{Z}_q$

$X_i = H_1(ID_i, L_i)$

$N_i = X_i \oplus D_i$ and $V_i = H_1(X_i, D_i)$

$\xleftarrow{\{N_i, V_i\}}$

Store $\{N_i, V_i, b_i\}$ in the mobile device

Mobile device holds

$\{N_i, V_i, b_i, n, q, a, \chi_\beta, H_1(.), H_2(.), H_3(.)\}$

Fig. 1 User registration phase of Islam and Basu's [3] PB-3PAKA protocol

1. *Initialization phase:*
   During initialization phase, the server S selects three one-way hash function, $H_1,\ H_2,\ H_3 :\ \{0, 1\}^* \rightarrow \{0, 1\}^l$ as well as public parameters $\{n, q, a, \chi_\beta\}$. Also, server S selects a secret key s, where $s \in \mathbb{Z}_q$.

2. *User registration phase:*
   In user registration phase, the user $\{A, B\}$ chooses a identity ID_i, password PW_i from the dictionary D and $b_i \in \mathbb{Z}_q$ and compute L_i. After that sends $\{ID_i, L_i, personal\ information\}$ to server S. On the server's side, server S verifies the personal information of the user $\{A, B\}$ and computes D_i, X_i, N_i and V_i.

3. *Authenticated key agreement phase:*
   In authenticated key agreement phase, the user $\{A, B\}$ computes his public keys x_A and x_B and the parameters Σ_A and Σ_B. $\{ID_A, T_A, x_A, \sigma_A\}$ and $\{ID_B, T_B, x_B, \sigma_B\}$ are sent to the server.
   On the server side, the server authenticates the user $\{A, B\}$ and sends its identity and parameters Σ_{S_A}, Σ_{S_B} to users A and B, respectively.
   User $\{A, B\}$ authenticates to the server and computes t_A, t_B as well as signal functions w_A, w_B. Lastly, User A and User B send messages to each other, authenticate each other, and finally generate a session key.

3 Cryptanalysis of Islam and Basu's Protocol

In this section, we describe the cryptanalysis of Islam and Basu's [3] protocol based on a password-based three-party authenticated key agreement protocol for mobile devices in post-quantum environment. After examining the protocol, we find that the protocol is vulnerable to dishonest user's attack and signal leakage attack. These attacks are described below as follows.

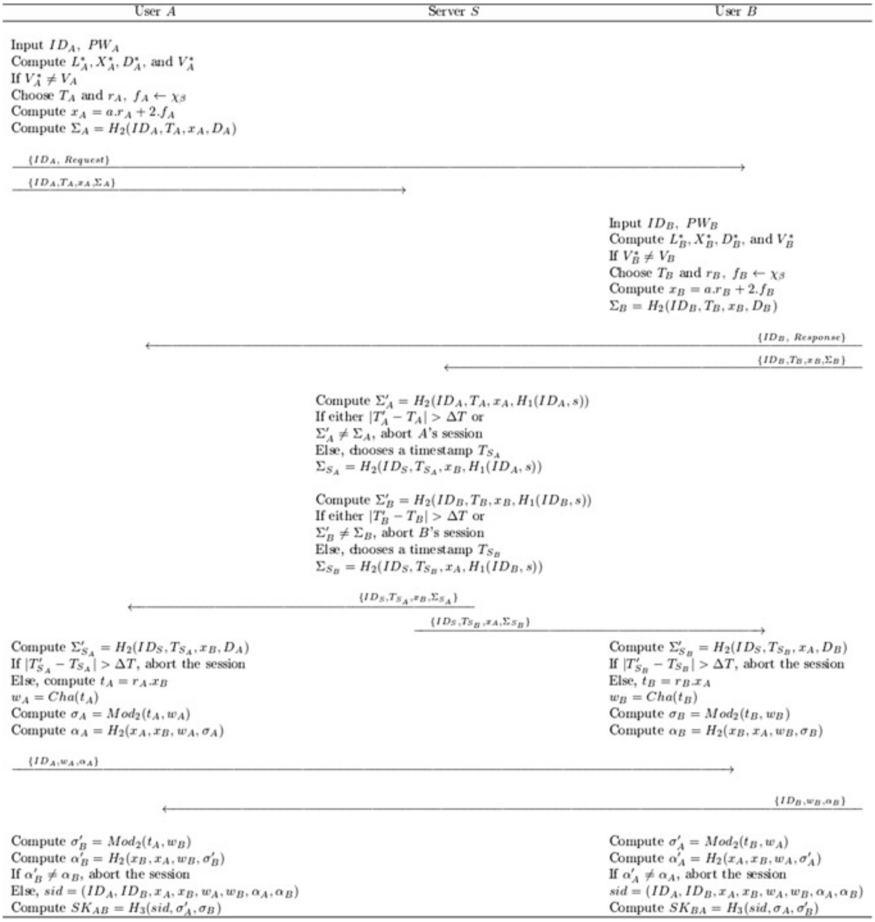

User A
Server S
User B
Input ID_A, PW_A
Compute L^*_A, X^*_A, D^*_A, and V^*_A
If $V^*_A \neq V_A$
Choose T_A and r_A, $f_A \leftarrow \chi_\beta$
Compute $x_A = a.r_A + 2.f_A$
Compute $\Sigma_A = H_2(ID_A, T_A, x_A, D_A)$
$\{ID_A, Request\}$
$\{ID_A, T_A, x_A \Sigma_A\}$
Input ID_B, PW_B
Compute L^*_B, X^*_B, D^*_B, and V^*_B
If $V^*_B \neq V_B$
Choose T_B and r_B, $f_B \leftarrow \chi_\beta$
Compute $x_B = a.r_B + 2.f_B$
$\Sigma_B = H_2(ID_B, T_B, x_B, D_B)$
$\{ID_B, Response\}$
$\{ID_B, T_B, x_B, \Sigma_B\}$
Compute $\Sigma'_A = H_2(ID_A, T_A, x_A, H_1(ID_A, s))$
If either $|T'_A - T_A| > \Delta T$ or
$\Sigma'_A \neq \Sigma_A$, abort A's session
Else, chooses a timestamp T_{S_A}
$\Sigma_{S_A} = H_2(ID_S, T_{S_A}, x_B, H_1(ID_A, s))$
Compute $\Sigma'_B = H_2(ID_B, T_B, x_B, H_1(ID_B, s))$
If either $|T'_B - T_B| > \Delta T$ or
$\Sigma'_B \neq \Sigma_B$, abort B's session
Else, chooses a timestamp T_{S_B}
$\Sigma_{S_B} = H_2(ID_S, T_{S_B}, x_A, H_1(ID_B, s))$
$\{ID_S, T_{S_A}, x_B, \Sigma_{S_A}\}$
$\{ID_S, T_{S_B}, x_A, \Sigma_{S_B}\}$
Compute $\Sigma'_{S_A} = H_2(ID_S, T_{S_A}, x_B, D_A)$
If $|T'_{S_A} - T_{S_A}| > \Delta T$, abort the session
Else, compute $t_A = r_A.x_B$
$w_A = Cha(t_A)$
Compute $\sigma_A = Mod_2(t_A, w_A)$
Compute $\alpha_A = H_2(x_A, x_B, w_A, \sigma_A)$
Compute $\Sigma'_{S_B} = H_2(ID_S, T_{S_B}, x_A, D_B)$
If $|T'_{S_B} - T_{S_B}| > \Delta T$, abort the session
Else, $t_B = r_B.x_A$
$w_B = Cha(t_B)$
Compute $\sigma_B = Mod_2(t_B, w_B)$
Compute $\alpha_B = H_2(x_B, x_A, w_B, \sigma_B)$
$\{ID_A, w_A, \alpha_A\}$
$\{ID_B, w_B, \alpha_B\}$
Compute $\sigma'_B = Mod_2(t_A, w_B)$
Compute $\alpha'_B = H_2(x_B, x_A, w_B, \sigma'_B)$
If $\alpha'_B \neq \alpha_B$, abort the session
Else, $sid = (ID_A, ID_B, x_A, x_B, w_A, w_B, \alpha_A, \alpha_B)$
Compute $SK_{AB} = H_3(sid, \sigma'_A, \sigma_B)$
Compute $\sigma'_A = Mod_2(t_B, w_A)$
Compute $\alpha'_A = H_2(x_A, x_B, w_A, \sigma'_A)$
If $\alpha'_A \neq \alpha_A$, abort the session
$sid = (ID_A, ID_B, x_A, x_B, w_A, w_B, \alpha_A, \alpha_B)$
Compute $SK_{BA} = H_3(sid, \sigma_A, \sigma'_B)$

Fig. 2 Authenticated key agreement phase of Islam and Basu's [3] PB-3PAKA protocol

3.1 Dishonest User's Attack

Dishonest user's attack is feasible due to the registration phase of Islam and Basu protocol (see Fig. 1 for a complete description of the registration phase). We show that the adversary correctly recovers the server's secret key s where $s \in \mathbb{Z}_q$. Here, there are two users and one server. The first is user A and the second is user B. We assume that either of these two users is an adversary (Eve). The following steps of dishonest user's attack are as follows:

Step 1: First of all, the adversary $\mathcal{A}$ chooses its ID_A and password PW_A and along with it also chooses a random element $b_A \in \mathbb{Z}_q$. Now, the adversary computes the parameter $L_A = H_1(PW_A, b_A)$ using the hash function on its password

PW_A and random element b_A (see Fig. 1). After this, $\mathcal{A}$ sends her $\{ID_A, L_A\}$, and personal information to the server through a secure network.

Step 2: The server receives the $\{ID_A, L_A\}$ and personal information of the adversary and verifies the adversary's personal information. Now, the server computes a parameter D_i using the hash function on his master secret key s where $s \in \mathbb{Z}_q$ and the adversary's ID_A. With this, the server computes a parameter X_i using the hash function on the adversary's ID_A and L_A.
Lastly, the server computes the parameters $V_i = H_1(X_i, D_i)$, $N_i = X_i \oplus D_i$ and sends these parameters V_i, N_i to the adversary.

Step 3: Now, the adversary has the knowledge of V_i, N_i as well as the value of $X_i = H_1(ID_A, L_A)$ because the server has computed the parameter X_i using the hash function on (ID_A, L_A) (see Fig. 1). Further, Adversary $\mathcal{A}$ can easily find the value of D_i by using X_i and N_i parameters.
In the protocol, the master secret key of server s belongs to $\mathbb{Z}_q$, and the adversary puts the value of s from 0 to $q - 1$ in $H_1(ID_i, s)$ to match the value of D_i. If adversary guesses the correct value of s, then adversary recovers the server's master secret key s.

3.2 Signal Leakage Attack and Improved Signal Leakage Attack

In TLS v1.3, the key exchange computations and communication costs are saved by reusing public and private key pairs. Resumption mode is used to establish the overwhelming majority of TLS connections in the real world. The security is compromised when keys are reused in TLS, due to this the PB-3PAKA [3] protocol is vulnerable to a signal leakage attack and improved signal leakage attack. Therefore, the adversary can retrieve the user's secret key (see Fig. 2 for a complete description of the signal leakage attack).

Attack overview:

Islam and Basu's PB-3PAKA protocol has two parties, A and B and one server S. As of TLS v1.3, we reuse the secret keys r_A and r_B, respectively, of both parties, A and B in the Islam and Basu's PB-3PAKA protocol. Suppose party A plays the role of an adversary (Eve) and party B is an honest party. Adversary wants to recover the secret key r_B of the honest party B. She generates her malicious public key x_A and sends it to party B. Party B computes $t_B = r_B.x_A$ and signal function w_B using the adversary's malicious public key and sends the signal function w_B to the adversary. So, the adversary retrieves the secret key r_B by observing the signal function w_B sent by the party B. For detailed description, see below attack.

Signal Leakage Attack:

Let the value of the adversary's secret key r_A is 0 and the value of adversary's error term f_A to be 1. By which the public key of the adversary will be $x_A = k$, where the value of $k \in \mathbb{Z}_q$. Now, the adversary sends its ID_A, public key x_A, and other parameters T_A (Timestamp), $\Sigma_A = H_2(ID_A, T_A, x_A, D_A)$ to the server.

Similarly, party B also derives its public key and other parameters and sends them to the server. Also, sends its $< ID_B\ and\ response >$ to the adversary.

Now, the server sends the public key of the adversary to party B and the public key of party B to the adversary.

After receiving the public key of the adversary sent by the server, party B computes t_B where $t_B = r_B.x_A$ (here, r_B is the secret key of party B). In addition, it also computes the signal function

$$w_B = Cha(t_B)$$

modular function

$$\sigma_B = Mod_2(t_B, w_B)$$

and the parameter $\alpha_B = H_2(x_B, x_A, w_B, \sigma_B)$ (see Fig. 2).

Now,

$$\begin{aligned} t_B[i] &= r_B.x_A[i] \\ &= r_B(a.r_A + k.f_A)[i] \\ &= k.r_B[i] \end{aligned}$$

where $k \in \{0, \ldots, q-1\}$.

As soon as the adversary will vary the value of k, likewise she will guess the value of $k.r_B[i]$ correctly, because the number of the signal w_B changes for every coefficient of $r_B[i]$. When there is a change in the signal for any ith coefficient of $r_B[i]$, then the number of that change is exactly $2.r_B[i]$. But the value of $+1$ and -1 only gives signal change of the same number, due to which the adversary can only guess the value up to ± sign. For the value of $-r_B$, the value of k changes in the reverse direction which is a multiple of r_B.

Therefore, to find out the exact value of the r_B coefficient, the adversary initiates the q number of sessions with party B with its public key (for more details, see [1]).

Improved Signal Leakage Attack:

Attack details:

In the beginning, adversary $\mathcal{A}$ sends its $< ID_A\ and\ request >$ to party B. Moreover, she derives her public key $x_A = a.r_A + k.f_A$, here r_A and f_A are the adversary's secret key and the error term, respectively (see Fig. 2). Now, two cases arise here. In the first case, the adversary takes the value of r_A as 0, and in the second case, the value of r_A is taken as very small depending on error distribution.

In the improved signal leakage attack case, the adversary chooses the value of error term f_A as 1 and selects the value of secret key r_A according to the error distribution. Therefore, the public key of the adversary is $x_A = a.r_A + k$ so that the public key of the adversary cannot be distinguished. Here,

$$\begin{aligned} t_B &= r_B.x_A \\ &= r_B(a.r_A + k.f_A) \\ &= a.r_A.r_B + k.r_B \end{aligned}$$

and signal function

$$\begin{aligned} w_B &= Cha(t_B) \\ &= Cha(a.r_A.r_B + k.r_B) \end{aligned}$$

As adversary $\mathcal{A}$ iterates over k values, $a.r_A.r_B$ remains constant.

Consequently, $\mathcal{A}$ continues to observe the signal changes of $r_B[i]$ while she varies the values of k toward the positive values and starts from $k = 0$. After this, the adversary records the first signal change in w_B at $k = k_1$.

The adversary then varies k toward the negative values and observes the first signal change in w_B and in this direction it records the first signal change at $k = k_2$.

Now, the period of region T or T^c in multiples of $r_B[i]$ is $k_1 - k_2$. The period of the signal change is $k_1 - k_2$, due to which the value of $r_B[i]$ up to the ± sign is revealed by $\frac{q}{2.(k_1 - k_2)}$. The process of changing the signal continues till the signal becomes stationary after the change, and then adversary can query a small constant number here more than $\frac{q}{2}$ times.

In this way, the adversary can recover $r_B[i]$ up to the sign by doing $\frac{q}{2} + c$ queries. Here c is a small value because as the value of k increases, the value stabilizes and $k.r_B[i]$ moves away from the boundary point. Now, adversary performs $q + c$ queries so that $\mathcal{A}$ can recover the exact value of the secret (for more details, see [2]).

4 Conclusion

We have studied Islam and Basu's [3] proposed protocol based on a password-based three-party authenticated key agreement protocol for mobile devices in post-quantum environment (PB-3PAKA). It has been found that their protocol is vulnerable to dishonest user's attack. The security is compromised when keys are reused in TLS v1.3, and due to this the PB-3PAKA protocol is vulnerable to signal leakage attack. In future, we will propose an improved protocol to overcome the above-identified attacks on Islam and Basu's proposed protocol.

References

1. Ding J, Alsayigh S, Saraswathy R, Fluhrer S, Lin X (2017) Leakage of signal function with reused keys in RLWE key exchange. In: 2017 IEEE international conference on communications (ICC). IEEE, pp 1–6
2. Ding J, Fluhrer S, Rv S (2018) Complete attack on RLWE key exchange with reused keys, without signal leakage. In: Australasian conference on information security and privacy. Springer, pp 467–486
3. Islam SH, Basu S (2021) PB-3PAKA: password-based three-party authenticated key agreement protocol for mobile devices in post-quantum environments. J Inf Secur Appl 63:103026
4. Kirkwood D, Lackey BC, McVey J, Motley M, Solinas JA, Tuller D (2015) Failure is not an option: standardization issues for post-quantum key agreement. In: Workshop on cybersecurity in a post-quantum world, p 21
5. Rescorla E (2018) The transport layer security (TLS) protocol version 1.3. Technical report