A Secure and Efficient Key Agreement Protocol Based on Certificateless Cryptography

Nashwa A.F. Mohamed, Mohsin H.A. Hashim and Eihab B.M. Bashier,
Faculty of Mathematical sciences, University of Khartoum, Khartoum, Sudan,

Mohamed E.H. Hassouna
Faculty of computer Studies National Ribat University, Khartoum, Sudan

Abstract

Almost all the certificateless two-party authenticated key agreement (CTAKA) protocols found in the literature, suffer either serious security problems or inefficient performance that involves high computational costs. In this paper, we design a secure and efficient CTAKA protocol. Within the proposed CTAKA protocol, the KGC publishes the public keys of the users in a public directory (LDAP server) that has a certificate to prove its identity to the users. The LDAP certificate is the only existing certificate within the scheme. Both the two communicating parties are able to compute a common secret per session key using a secure generated random number. The protocol is fully secure against type 1 attack and fully secure against type 2 attacks under the assumptions that the KGC is an honest party and each party within the network has the full capability to keep its secret values safe. Moreover, the setup of the protocol does not include pairings and the whole key agreement process requires only four point scalar multiplications, two point additions, one hash function evaluation and one message exchange.

1. Introduction

Key agreement protocols are fundamental primitives of cryptography. A key agreement protocol allows two or more parties to securely establish secret keys in the presence of an adversary. Among all key agreement protocols the attractive one is the authenticated key agreement protocol, since it offers the assurance that only the participating parties of the protocol can compute the agreed key.

In 2003 Al-Riyami and Paterson [1] proposed the first certificateless authenticated two-party key agreement protocol (CTAKA). In 2006 Mandt [2] pointed that Al-Riyamy and Paterson's [1] CTAKA protocol does not have resistance to leakage of ephemeral keys. He proposed an alternative CTAKA protocol. Mandt showed that his protocol can be used to establish keys between users of different key generation centers.

An interesting piece of research in certificateless key agreement protocols, was introduced by, Swanson [3] in his master's thesis. He introduced the first formal security model for certificateless authenticated key agreement protocols. Swanson's model was an extended version of the extended model introduced by Cantti and Krawczyk (eCK) [4] for traditional authenticated key exchange. Swanson turned the table on most existing certificateless key agreement protocols that time, when he examined Al-Riyami and Paterson [1], Mandt [2], Wange et al [5], Shao [6] and Shi and Lee [7] certificateless authenticated two-party key agreement protocols in his model and pointed that all of them allow for practical attacks of varying severity, ranging from lack of resistance to leakage of ephemeral key up to a man-in-the-middle attack.

Based on Swanson's model [3], Lippold et al. [8] proposed in 2009 a strong formal model for secure certificateless authenticated key agreement protocols. Their proposed protocol was the first one-round protocol with a security proof in random oracle model for certificateless key agreement. It fulfilled all notions of security of their model and withstands recent attacks on certificateless key agreement protocols. The authors also pointed that Swanson's model was weak because it assumed that after the adversary replaces a public key of some party, the model allows that party to continue using its original public key (and its matching private key) to make its computations.

The major problem with the Lippold et al's protocol [8] was that it requires computation of 5 exponentiations and 10 parings which is an unacceptable, computational cost (since the relative computation cost of a pairing is approximately twenty times higher than that of a scalar multiplication over elliptic curve group [9]).

In 2010 Zhang et al [10] pointed that none protocol of [[1], [2], [7]] has been proved secure with a formal proof, and most of them suffer from heavy pairing computation during the establishment of the session key

phase. Zhang et al. [10] proposed a security model for authenticated key agreement (AKA) protocols using certificateless cryptography. Then, they proposed a simulatable CTAKA protocol that requires only one pairing operation.

In 2011, Mokhtarnameh et al. [11], proposed an authenticated key agreement protocol. Then, Yang et al [12] stated that the Mokhtarnameh et al. [11] (MHM) protocol was insecure against man-in-the-middle attack. They proposed a modified protocol based on MHM protocol and claimed that their protocol was secure. Unfortunately neither the MHM nor Yang et al protocols were secure. The authors of the two protocols could not notice that their protocols suffer from key escrow problem. In MHM protocol[11], the KGC can simply multiply the public term P_A by its secret s ($sP_A = sx_AQ_A = x_AsQ_A = x_AD_A = S_A$) to obtain the full secret key of entity A, whereas, in Yang et al.'s protocol[12], the KGC can simply multiply the public key term Y_A by its secret s to obtain the full private key of user A ($sY_A = sx_AQ_A = x_AsQ_A = x_AD_A = S_A$).

All the above mentioned certificateless two-party authenticated key agreement protocols were from bilinear pairings.

As noted before, the relative computation cost of a pairing is approximately twenty times higher than that of a scalar multiplication over elliptic curve group [9]. To improve the performance, several CTAKA protocols without pairings have been proposed in the literature. In 2009, Geng and Zhang [13] and Hou and Xu [14] proposed CTAKA protocols without pairings. In 2011, Yang and Tan [15] proposed a provably secure pairing-free certificateless authenticated key agreement protocol along with a new model for CTAKA. He et al [16] also proposed a CTAKA protocol without pairing. Unfortunately, their protocol was vulnerable to the type 1 adversary as shown in [17].

In [19] He and Chen proposed a new CTAKA protocol without pairings. In their protocol, the user computes only five elliptic curve scalar multiplications for the key agreement. They also proved that their protocol was secure under the random oracle model.

In [20] He and Padhye, proposed a new CTAKA protocol and they proved its security in the eCK model.

In this paper, we propose a certificateless two-parties authenticated key agreement protocol in which the KGC publishes the public keys of the users in a public directory (LDAP server) that has a certificate to prove its identity. The LDAP certificate is the only existing certificate within the scheme. The KGC is in an offline connection with the LDAP server and the users do not have direct connection with the KGC. Within the scheme, the first communicating party downloads the public key of the second one from the public directory, generates a random number, encrypts it with the public key of the second party and sends this encrypted number to the second party among the hello message. The second party downloads the public key of the first party and decrypts the encrypted random number using its private key. Then, each of the parties uses its own secret key together with the public key of the other party and the random number generated by the first party to generate a per session symmetric key without interactions between them except the initial communication. The protocol introduces solutions to the man-in-the-middle attack, key escrow problem and it provides many other security properties that make the protocol fully secure, under the assumption that that the KGC is an honest party and each party has the full capability to protect its secret values. In addition to that, the proposed scheme is a level 3 secure system.

The rest of this paper is organized as follows. In Section 2 we review in the certificateless public key cryptography as introduced by Al-Riyami and Paterson [1], and then we propose the modified CL-PKC. We present the proposed certificateless two-party authenticated key agreement protocol in Section 3. Efficiency comparison with other pairing free CTAKA protocols is presented in Section 4. In Section 5, we state the security properties of the proposed protocol. Finally, we conclude the paper in Section 6.

2. Certificateless public key cryptography

In 2003 Al-Riyami and Paterson [1] introduced the concept of Certificateless Public Key Cryptography (CL-PKC) to overcome the key escrow limitation of the identity-based public key cryptography (ID-PKC). In CL-PKC a trusted third party called Key Generation Center (KGC) supplies a user with a partial private key. Then, the user combines the partial private key with a secret value (that is unknown to the KGC) to obtain his full private key. In this way the KGC does not know the users private keys. Then the user combines his secret value with the KGC's public parameters to compute his public key.

Compared to the ID-PKC, the trust assumptions made of the trusted third party in CL-PKC are much reduced. In IDPKC, users must trust the private key generator (PKG) not to abuse its knowledge of private keys in performing passive attacks, while in CL-PKC, users need only trust the KGC not to actively propagate false public keys [1].

In CL-PKC a user can generate more than one pair of key (private and public) for the same partial private key. To guarantee that KGC does not replace a user's public key Al-Riyami and Paterson[1] introduced a binding technique to bind a user's public key with his private key. In their binding scheme, the user first fixes his secret value and his public key and supplies the

KGC his public key. Then the KGC redefine the identity of the user to be the user's identity concatenated with his public key. By this binding scheme the KGC replacement of a public key apparent, and equivalent to a certificate authority forging a certificate in a traditional PKI.

2.1. Al-Riyami and Paterson scheme

In this subsection we give a general description to Setup, Set-Secret-Value, Partial-Private-Key-Extract, Set-Private-Key and Set-Public-Key algorithms as introduced by Alriyami and Paterson [1]. Below, we state the algorithms that were presented by Alriyami and Paterson [1].

Let k be a security parameter given to the Setup algorithm and Ω be a Bilinear Diffie-Hellman Problem (BDH) parameter generator with input k.

- **Setup (running by the KGC):** this algorithm runs as follows:
  i) Run Ω on input k to generate output $<G_1, G_2, e>$, where $G_1$and G_2 are groups of some order q and $e : G_1 \times G_1 \rightarrow G_2$ is a pairing.
  ii) Choose an arbitrary generator $P \in G_1$.
  iii) Select a master-key s uniformly at random from Z^*_q and set $P_0 = sP$.
  iv) Choose cryptographic hash functions $H_1 :\{0,1\}^* \rightarrow G_1^*$ and $H_2 : G_2 \rightarrow \{0,1\}^n$ where n is the bit-length of plaintexts taken from some message space $M=\{0,1\}^n$ with a corresponding ciphertext space $C = G_1 \times \{0,1\}^n$.

  Then, the KGC publishes the system parameters $params =<G_1, G_2, e, n, P, P_0, H_1, H_2>$, while the secret master-key s is saved secure by the KGC.

- **Set-Secret-Value (running by the user):**
  This algorithm takes as inputs params and client A's identifier ID_A as inputs. Client A selects $x_A \in Z^*_q$ at random and output x_A as A's secret value. Then, he/she computes $X_A = x_AP$ and sends X_A to the KGC.

- **Partial-Private-Key-Extract (running by the KGC):**
  This algorithm takes as input an identifier $ID_A \in \{0,1\}^*$ and X_A, and carries out the following steps to construct the partial private key for client A with identifier ID_A.
  (a) Compute $Q_A = H_1(ID_A||X_A)$.
  (b) Output the partial private key $D_A = sQ_A \in G_1$.

  Client A when armed with it's partial private key D_A, he/she can verify the correctness of the partial private key D_A by checking $e(D_A, P) = e(Q_A, P_0)$.

- **Set-Private-Key (running by the user):**
  This algorithm takes as inputs params, an client A's partial private key D_A and A's secret value $x_A \in Z^*_q$. Client A transforms partial private key D_A to private key S_A by computing $S_A = x_AD_A = x_AsQ_A \in G_1$.

- **Set-Public-Key (running by the user):**
  This algorithm takes as inputs params and client A's secrete value $x_A \in Z^*_q$ as inputs and constructs A's public key as $P_A =<X_A, Y_A >$, where $X_A = x_AP$ and $Y_A = x_AP_0 =x_AsP$.

Both client A and client B can verify that the same KGC master key has been used in each other's public keys. A checks if $e(X_B, P_0) = e(Y_B, P)$ and B checks if $e(X_A, P_0) =e(Y_A, P)$.

2.2. The modified CL-PKC algorithm

Our purpose in this section is to introduce a modified CL-PKC algorithm which provides a certificateless two-party authenticated key agreement protocol (CTAKA) without interactions.

To do this, we make modifications in the binding scheme introduced by Alriyami and Paterson [1]. In the proposed algorithm; the Setup, Set-Secret-Value, Partial-Private-Key-Extract and Set-Private-Key algorithms are taken from the original algorithm presented by Alriyami and Paterson [1], while we make modifications in the Setup, Set-Secret-Value and Set-Public-Key algorithms. Below is the description of the modified algorithm.

- **Setup (running by the KGC):**
  the KGC chooses a secret parameter k to generate G_1, G_2, P, e where $G_1$and G_2 are two groups of a prime order q, P is a generator of G_1 and $e : G_1 \times G_1 \times G_2$ is a bilinear map. The KGC randomly generates the system's master key $s \in Z^*_q$ and computes the system public key $P_{pub} = sP$. Then, the KGC chooses cryptographic hash functions H_1 and H_2, where $H_1 : \{0,1\}^* \times G_1 \rightarrow G_1$ and $H_2 : G_1 \times G_1 \times G_1 \rightarrow \{0,1\}^n$. Finally, the KGC publishes the

system parameters *params*=<G_1, G_2, e, P, $Ppub$, H_1, H_2, n>, while the secret master-key is saved and secured by the KGC.

- **Set-Secret-Value (running by the user):** the client A with the identity ID_A downloads the system's public parameters from the KGC. Then, client A picks two random secret values x_A, $x'_A \in Z^*_q$. Then, he/she computes $X_A = x_A x'_A P$ and sends X_A to the KGC.

- **Partial-Private-Key-Extract (running by the KGC):** on receiving X_A computed by client A with identity ID_A, the KGC first computes $Q_A = H_1(ID_A \| X_A)$, then it generates the partial private key of client A as $D_A = sQ_A$.
  Client A when armed with it's partial private key D_A, he/she can verify the correctness of the partial private key D_A by checking $e(D_A, P) = e(Q_A, P_0)$.

- **Set-Private-Key (running by the user):** when client A receives D_A from the KGC, he/she computes his full private key $S_A = x_A D_A$.

- **Set-Public-Key (running by the user):** the client A with identity ID_A computes $Q_A = H_1(ID_A \| X_A)$, $Y_A = x_A x'_A Q_A$ and sets < X_A, Y_A > as his long-term public key P_A. Finally, user A sends Y_A to the KGC.

Both clients A and B can verify that the same KGC master key has been used in each other's public keys. Client A checks if $e(X_B,Q_B) = e(P,Y_B)$ and client B checks if $e(X_A,Q_A) = e(P, Y_A)$.

The purpose of the secret value x'_A is to prevent the key escrow problem that can be performed by the KGC. That is if the secret value x'_A is not a part from Y_A (i.e. $Y_A = x_A Q_A$ as Proposed by Yang et al [12]), the KGC can simply multiply Y_A by its secret s to obtain the full private key of user A.

3. The proposed CTAKA

In this section, we first describe the CTAKA protocols proposed by Mohamed et al. in [18], then we introduce an improved CTAKA protocol.

3.1. Mohamed et al. [18] CTAKA protocol

Assuming that client A has a private key $S_A = x_A D_A$, a public key P_A =< X_A, Y_A > and client B has a private key $S_B = x_B D_B$, a public key P_B=<X_B, Y_B >. Then, the common authenticated per session secret key can be computed at both sides as follow:

- Client A sends Hello message to client B.
- Client B requests A's public key from the KGC. If client A within the domain, the KGC generates ephemeral random number $t \in Z^*_q$ and sends t and A's public key to client B, otherwise, KGC sends error message.
- Client B replies to client A by rejection or sends hello message.
- Client A requests B's public key from the KGC, then the KGC sends t and B's public key to client A.
- Client A computes the secret key $K_{AB} = tx_A x'_A X_B$, whereas client B computes the secrete key $K_{BA} = tx_B x'_B X_A$.
- Client A computes the shared key $K_A = H_2(Q_A, Q_B, K_{AB})$, whereas client B computes the shared key $K_B = H_2(Q_A, Q_B, K_{BA})$.

It can easily seen that $K_A = K_B$ since $K_{AB} = tx_A x'_A x X_B = tx_A x'_A x_B x'_B P = tx_B x'_B x_A x'_A P = tx_B x'_B X_A = K_{BA}$.

3.2. The improved protocol

Involving the KGC in each session is not practical and can make the protocol vulnerable to denial of service attack. Therefore, we propose the following improved algorithm:

- Client A downloads the public key of client B from the public directory. He/she generates a random number a number $t \in Z^*_q$, encrypts it using the public key of client B. Then, sends a Hello message to client B including the encrypted number t.
- Client A computes the secrete key $K_{AB} = tx_A x'_A X_B$, then the shared key $K_A = H_2(Q_A, Q_B, K_{AB})$.
- On receiving the Hello message of client A, client B downloads A's public key from the public directory, then he/she decrypts the Hello message to obtain the secret number t.
- Client B computes the secrete key $K_{BA} = tx_B x'_B X_A$, then computes the shared key $K_B = H_2(Q_A, Q_B, K_{BA})$.

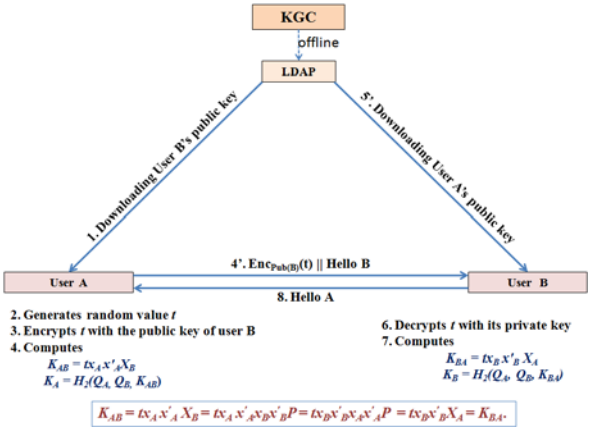

Figure 1. The proposed certificateless authenticated two party key agreement protocol

4. Efficiency Comparison with Other Pairing-free CTAKA Protocols

In this section we prove that our protocol is an efficient one compared with other pairing-free CTAKA protocols , i.e. Geng et al.'s protocol [13], Hou et al.'s protocol [14], Yang et al.'s protocol [15], He et al.'s protocols [16], [17], He and Padhye's protocol [20] and Mohamed et al. [18] protocol, Table 2 blow shows the comparison between pairing-free CTAKA protocols in terms of computational cost and number of message exchanges between parties, we follow He and Padhye [20] in defining some notations as follows.

- T_{mul}: The time of executing a scalar multiplication operation of point.
- T_{add}: The time of executing an addition operation of point.
- T_{inv}: The time of executing a modular inversion operation.
- T_h: The time of executing a one-way hash function.

Table 1. Computational cost and needed number for message exchanges in different pairing-free CTAKA protocols.

Protocol	Computational Cost	Message Exchange
Geng et al.'s [13]	$7T_{mul} + 2T_h$	2
Hou et al.'s [14]	$6T_{mul} + 2T_h$	2
Yang et al.'s [15]	$9T_{mul} + 2T_h$	2
He et al's [16]	$5T_{mul} + 3T_{add} + T_{inv} + 2T_h$	3
He et al's [17]	$5T_{mul} + 4T_{add} + 2T_h$	2
He and Padhye [20]	$5T_{mul} + 3T_{add} + 2T_h$	2
Mohamed et al. [18]	$1\ T_{mul} + 1\ T_h$	0
Improved CTAKA	$4T_{mul} + 2T_{add} + 1T_h$	1

5. Security properties of our proposed schemes

In this section, we state some of the security properties that are provided by the proposed scheme. We follow Swanson and Jao [3] in the statement of some of these properties in terms of our scheme.

- **Long-term binding public key with corresponding private key:** the long term public key $P_A = (X_A, Y_A)$ for a client A with identity ID_A is related to the partial private key D_A, since $D_A = sH_1(ID_A||X_A)$. Therefore, there is one-to-one correspondence between the public key and the partial private key of either of the two communicating parties, and insures that each user can create only one long term public key for the corresponding private key. The existence of two valid public keys for the same identity guarantees that the KGC will be identified to misbehave in issuing both corresponding partial private keys.
- **Key agreement without interactions:** the most probable attack during the run of a key agreement protocol is the man-in-the-middle attack. Our proposed key agreement provides resistance against the man-in-the middle attack, by enabling either of the two communicating parties to compute the shared secret key using its own secrete values, the other party's public key and a randomly generated number, without any interaction between them except the hello message.
- **Authentication:** since the shared per-session secret key is generated using the client's secrete values and the other client's public key and public parameter, then authentication of entities are provided.
- **Known key security:** Each session key is unique, because both the two communicating parties make use of a random number t which is generated randomly in each protocol run, thus the knowledge of previous session keys (if it happened to be) does not help an adversary to derive information about other session keys.
- **Unknown key share resilience:** The public parameters Q_A and Q_B are included in the computation of the common secret key. Therefore, entities A and B know who they share the key with.
- **Key-compromise impersonation resilience:** An adversary who has compromised the long-term private key of a client A is unable to

compute the session key, because x_A and x'_A is also required for computing the session key.

- **Weak perfect forward secrecy:** Suppose that an outside adversary has compromised long-term secret keys S_A; S_B; x_A; x'_A; x_B; x'_B; D_A and D_B. He still cannot obtain the secret random number t, because these long-term secret keys are unrelated to the random number t, thus the adversary is unable to determine previously established session keys. The KGC (if assumed entrusted) can compute the per session key if it compromised long-term secret keys of either clients.
- **Key control:** Neither entity should be able to force the session key to a preselected value.

6. Conclusions and remarks

In this paper, we introduced a certificateless two-party authenticated key agreement protocol; The KGC publishes the system parameters and public keys in a public directory. The public directory (LDAP server) possesses a certificate to prove its identity to both communicating parties. The communications between the KGC and the LDAP server is offline.

Both the two communicating parties are able to compute the same secret symmetric key without messages exchanges, except the first hello message, which is encrypted. This makes it impossible to carry out a man in the middle attack to obtain information about the encryption/decryption key. Moreover, the scheme is secure against known key attack, resilient against unknown key share and key-compromise impersonation, and secure against weak perfect forward secrecy.

Compared to the CTAKA protocol presented by Yang et. al (Yang et al., 2011) and the MHM protocol (Mokhtarnameh et al., 2011), the proposed scheme in this paper is resistant to the key escrow problem through using the second secret value x'_A. This comes in contrast to the scheme proposed by Yang et. al in(Yang et al., 2011) where the KGC can easily compute the full private key for user A by multiplying the public term Y_A by its master secret value s and also in contrast to the scheme proposed by Mokhtarnameh et al. (Mokhtarnameh et al., 2011), in which the KGC can easily compute the full private key for client A by multiplying the public term P_A by its master secret value s.

The proposed protocol is immune against any inside (type 2) or outside (type 1) attacks. Therefore, the CTAKA protocol introduced in this paper is fully secure, if we assume that the KGC is honest and each party within the network has the full capability to fully protect its secret values. On the other hand, The results explained in Table 1 show that all the pairing free protocols stated in the table have costs of at least five points multiplications, two hash function evaluations and two message exchanges, except Mohamed et al. [18] and the improved CTAKA protocol. The proposed protocol requires more point multiplications and hash function evaluations than the protocol of Mohamed et al. But it is much practical as it does not require the involvement of the KGC at each session initiation.

7. References

[1] S. Al-Riyami and K. Paterson, "Certificateless public key cryptography", in Asiacrypt 2003, ser. Lecture Notes in Computer Science, C. Laih, Ed., 2003, pp. 452–473, full version available at Cryptology ePrint Archive.

[2] T. K. Mandt, "Certificateless authenticated two-party key agreement protocols," master's thesis, Gjøvik University College, Department of Computer Science and Media Technology, 2006.

[3] C. Swanson, "*Security in key agreement: Two-party certificateless schemes*", 2008, master Thesis, University of Waterloo.

[4] B. A. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange", in Provable Security, ser. *Lecture Notes in Computer Science*, vol. 4784. Springer Berlin / Heidelberg, 2007, pp. 1–16.

[5] S. Wang, Z. Cao, and L. Wang, "Efficient certificateless authenticated key agreement protocol from pairings", *Wuhan University Journal of Natural Sciences*, vol. 11, no. 6, 2006.

[6] Z. hua Shao, "Efficient authenticated key agreement protocol using self-certified public keys from pairings", *Wuhan University Journal of Natural Sciences*, vol. 10, no. 1, pp. 267–270, 2005.

[7] Y. Shi and J. Li, "Two-party authenticated key agreement in certificateless public key cryptography", *Wuhan University Journal of Natural Sciences*, vol. 12, no. 1, pp. 71–74, 2007.

[8] G. Lippold, C. Boyd, and J. M. G. Nieto, "Strongly secure certificateless key agreement", *in Pairing, ser. Lecture Notes in Computer Science*, vol. 5671. Springer-Verlag, 2009, pp. 206–230.

[9] L. Chen, Z. Cheng, and N. P. Smart, "Identity-based key agreement protocols from pairings", *Int. J. Inf. Sec.*, vol. 6, no. 4, pp. 213–241, 2007.

[10] L. Zhang, F. Zhang, Q. Wu, and J. Domingo-Ferrer, "Simulatable certificateless two-party authenticated key agreement protocol", *Inf. Sci.*, vol. 180, pp. 1020–1030, March 2010.

[11] R. Mokhtarnameh, S. Ho, and N. Muthuvelu, "An enhanced certificateless authenticated key agreement protocol," *in Proc. of the 13th International Conference on Advanced Communication Technology(ICACT)*, 2011, pp. 802–806.

[12] H. Yang, Y. Zhang, and Y. Zhou, "*An improved certificateless authenticated key agreement protocol*", Cryptology ePrint Archive, Report 2011/653, 2011, http://eprint.iacr.org/ (Accessed 15/02/2012).

[13] M. Geng and F. Zhang, "Provably secure certificateless two-party authenticated key agreement protocol without pairing", *in Proceedings of the 2009 International Conference on Computational Intelligence and Security,* Volume 02, ser. CIS '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 208–212.

[14] M. Hou and Q. Xu, "A two-party certificateless authenticated key agreement protocol without pairing", *2nd IEEE International Conference on Computer Science and Information Technology*, pp. 412–416, 2009.

[15] G. Yang and C.-H. Tan, "Strongly secure certificateless key exchange without pairing", *in Proceedings of the 6th ACM Symposium on Information*, Computer and Communications Security, ser. ASIACCS '11. New York, NY, USA: ACM, 2011, pp. 71–79.

[16] D. He, J. Chen, and J. Hu, "A pairing-free certificateless authenticated key agreement protocol," International Journal of Communication Systems, 2011, (In press) DOI: 10.1002/dac.1265.

[17] D. He, Y. Chen, J. Chen, R. Zhang, and W. Han, "*A new two-round certificateless authenticated key agreement protocol without bilinear pairings*", Mathematical and Computer Modelling, 2011.

[18] N. Mohamed, M. Hassouna, and E. Bashier. "An almost fully secure pairing free certificateless key agreement protocol". *In Proceedings of the World Congress on Internet Security*, 2012, 2012. to appear.

[19] D. He and Y. Chen, "An efficient certificateless authenticated key agreement protocol without bilinear pairings," CoRR, vol. abs/1106.3898, 2011.

[20] Debiao He J. C., Padhye S., "*An efficient certificateless authenticated key agreement scheme*", Cryptology ePrint Archive, Report 2011/478, 2011, http://eprint.iacr.org/ (Accessed 15/02/2012).

8. Acknowledgement

The authors of this paper are pleased to acknowledge the full funding of the University of Khartoum and the Nile Center for Technology Research (NCTR) (http://www.nctr.sd) to the research led to this paper.

