Article can be accessed online at http://www.publishingindia.com

Analysis and Verification of a Key Agreement Protocol over Cloud Computing Using Scyther Tool

Hazem A. Elbaz*, Mohammed H. Abd-elaziz, Taymoor M. Nazmy*****

Abstract

Most of the cloud computing authentication mechanisms use public key infrastructure (PKI). Hierarchical Identity Based Cryptography (HIBC) has several advantages that well align with the demands of cloud computing. The main objectives of cloud computing authentication protocols are security and efficiency. In this paper, we clarify Hierarchical Identity Based Authentication Key Agreement (HIB-AKA) protocol, providing lightweight key management approach for cloud computing users. Then, we analyse the security properties of HIB-AKA protocol. We also show, a HIB-AKA security protocol proof using formal automated security analysis Scyther tool.

Keywords: Cloud Security, Security Analysis, Key Management, Hierarchical Identity-Based Authentication, Securtiy and Verifying Tools

Introduction

Security protocols play important role with wide use in many applications nowadays. The cryptographic key management systems are managed and protected throughout their life cycles by cryptographic techniques using cryptographic keys. Exchanging or distributing secret keys between two or more participants who want to establish a secure communication over an insecure channel is one of the most important roles of cryptographic keys. For this objective, key agreement protocols are widely used. Key agreement protocols allow two or more entities to establish a shared secret key to use in securing subsequent communication over an insecure channel (Farash *et al*., 2013; Martin, 2008). Security and efficiency are aspects taken into consideration to design the key agreement protocols. A secret key agreed between parties should be not disclosed to any entity which is an important aspect of security. An efficiency aspect should be taken care about optimum of computational and communication costs of the key agreement protocol (Chatterjee & Sarkar, 2011). Communicating over insecure channel, between two users needs authentication key agreement protocol to create a shared secret key to be guaranteed that they are indeed sharing this secret key with each other. Later, a large number of research has been made on identity based authenticated key agreement protocol, because of simplicity of public key management. Bilinear pairings on elliptic curves mostly use identity based two parties key agreement schemes as proposed in (Elbaz *et al*., 2014; Cao *et al*., 2010). According to Sanjit Chatterjee & Palash Sarkar (2011), each of two parties in the system using identity based cryptography, has its own private key and public key of each other, to calculate the secret shared key between them.

In this paper, we aim to analyse and verify the hierarchal identity-based authenticated key agreement (HIB-AKA) protocol. We will use the formal automated security analysis Scyther tool. There are many researches on verification of security protocols, therefor we will summarise some related works on security verifier tools as well as formalized and verification of privacy properties.

There are many tools for verifying and specifying security protocols such as AVISPA, ProVerif, SeVe , CASRUL or Scyther. Most of these tools focus on authentication and

* Faculty of Computer Science and Information System,Ain-Shams University, Cairo, Egypt. Email:hazem.baz@gmail.com
** Faculty of Computer Science and Information System,Ain-Shams University, Cairo, Egypt. Email:mhashem100@yahoo.com
*** Faculty of Computer Science and Information System,Ain-Shams University, Cairo, Egypt. Email:ntaymoor19600@gmail.com }.com

secrecy properties (Luu *et al*., 2012; Cheval & Blanchet, 2013; Basin *et al*., 2013). The AVISPA project aims at developing a push-button, industrial-strength technology for the analysis of large-scale Internet security-sensitive protocols and applications. AVISPA can use four methods in order to check a given security protocol: 1. OFMC (On the Fly Mode Checker) which uses symbolic techniques. 2. CL-AtSe that uses simplification heuristics and redundancy elimination techniques. 3. SATMC (SAT based Model Checker) that uses SAT-solvers in order to find a proposition leading to a fail in the model. 4. TA4SP that builds regular grammar in order to interpret and evaluate the intruder knowledge. There exist two different modes that can be used in AVISPA: Basic and Expert modes (Farash *et al*., 2014; Alegría *et al*., 2014). CASRUL manages the knowledge of principlesand checks if protocol is runnable. Itis a system for automatically verifying cryptographic protocols. Its outputis a set of rewrite rules describing the protocol itself, the goal to achieve the strategy of an intruder. CASRUL aims to model behaviour of intrude and permits to handle parallel sessions and composition of keys (Cortier & Warinschi, 2005; Modersheim, 2009). ProVerif is a fully automatic, efficient protocol that can handle an unbounded number of sessions, an unbounded message space, and any cryptographic primitives that can be represented by an equation theory and/or rewrite rules. Even if it does not always terminate, it was shown very efficient for many case studies (Cheval & Blanchet, 2013; Blanchet, 2011). Scyther is a tool for automatic verification of security protocols. The Dolev-Yao intrude model is used in Scyther analyses protocols. Its algorithm is guarantee to terminate, at which point Scyther establishes unbounded verification or bounded verification of a wide range of basic authentication and secrecy properties. Scyther has been deployed to analyse the standards of industrial security protocols IKE (v1 & v2) and ISO/IEC 9798. It is used to verify more advanced security properties of authenticated key exchange security models (Cremers & Mauw, 2012; Basin *et al*., 2014). The rest of the paper is organised as follows. The next section briefly explains the hierarchical identity-based key management and the corresponding concepts (bilinear pairing and the associated computational problems). The third section gives details on HIB-AKA protocol and the security analysis of properties desired for our proposed authenticated key agreement protocol. The fourth section shows our proposed protocol is probably secure using a formal security protocol verification Scyther Tool. Finally, a conclusion is made in the fifth section.

Background of Hierarchical Identity-Base Management

Bilinear Pairing

In this section, we describe bilinear pairings and their properties. More details can be found in papers by Chatterjee & Sarkar (2011) and Boneh & Franklin (2003).

Let G1 and G2 denote two groups of prime order q. G1 is an additive group and G2 a multiplicative group. Let P be a generator of G1. A pairing is a computable bilinear map between these two groups. Two pairings have been studied for cryptographic use, namely the Weil pairing and the Tate pairing.

For our purpose, let ê denote a general bilinear map ê: G1 ×G1 → G2, which satisfies the following three properties:

- Bilinear: If P,Q∈ G1 and a,b∈Z^*_q, then ê(aP,bQ) = ê(P,Q)^{ab}.
- Non-degenerative: There exist non-trivial points P,Q∈ G1 both of order q such that ê(P,Q)≠1.
- Computable: If P,Q∈ G1, ê (P,Q) ∈ G2 is efficiently computable (in polynomial time).

We say that G1 is a bilinear group if the group action in G1 can be computed efficiently and there exists a group G2 and an efficiently computable bilinear map ê : G1×G1 →G2 as above. Weil and Tate pairings associated with super singular elliptic curves or Abelian varieties can be modified in order to create such bilinear maps.

Computational Problems

Many pairing-based cryptographic protocols based on the hardness of the BDHP (Bilinear Diffie-Hellman Problem) for their security (Modersheim, 2009; Luu *et al*., 2012). Some computational problems related to the elliptic curve cryptography are:

- **Bilinear Diffie-Hellman Problem (BDHP):** Let G1 and G2 be two groups of prime order q. Let ê : G1 × G1 → G2, be a bilinear map and let P be a generator of G1. The BDH problem in (G1,G2,ê) is defined as: Given (P,xP,yP,zP) ∈ G1 for some x,y,z chosen at random from Z^*_q, compute ê(P,P)^{xyz}∈G2.
- **Discrete Logarithm Problem (DLP):**Given P,Q∈ G1, find an integer n such that P = nQ.

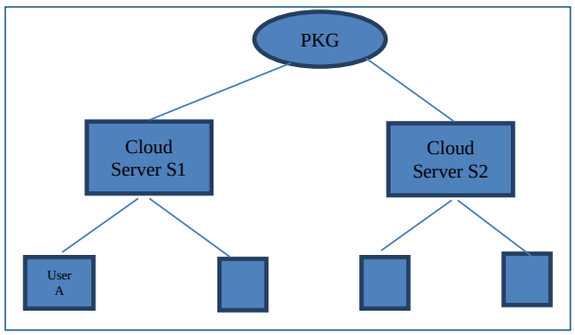
- **Computational Diffie-Hellman Problem (CDHP):** Given a tuple (P,aP,bP) ∈ G1 for a,b∈Z^*_q, find the element abP.

Hierarchal Identity based Key Management

The hierarchical identity-based key management scheme is composed of three levels using authenticated key agreement protocol. The top level is root PKG. The level-1 is domain PKGs, which are the cloud services in the cloud computing. The level-2 is users in the cloud computing. The user's public key consists of their identity and their domain's identity. For example, the cloud service identity is ID1, the user M's identity is IDm, and the identity of user M in hierarchical key management system is ID1 || IDm (Martin, 2008). For example, root PKG creates identity ID_Uni to a private cloud of a University. Identities of all users and servers in a private cloud or public cloud is managed and allocated by using sub-domain PKG. A hierarchal identity is created for user and server, which combines both identity of the user or server and the identity of the sub- domain. For example, the identity of email server in the private cloud of a University can be ID_Uni.email_server.

Figure 1 shows the hierarchical PKGs architecture in cloud computing.

Figure 1: Hierarchical PKGs Architecture in Cloud Computing



Key Generation

Root Setup: The root PKG operates as follows:

Step 1: Generate two cyclic groups G1, G2 of large prime order q and bilinear map e:G1×G1→G2, choose an arbitrary generator P0 ∈ G1;

Step 2: Root PKG picks a random s0 Z^*_q and sets Q0=s0P0 .Choose two hash functions: H1:{0,1}* → G1, H2:G2 → $\{0,1\}^n$. The root PGK's secret is s0.The system public parameters are (G1, G2, e, P0, Q0, H1, H2).

Lower-level setup: level-1 set up. The level-1 is cloud services. The cloud server S1 is the domain PKG. The cloud server S1 identity is ID1, then it picks a random s1 Z^*_q and keeps it secret.To obtain the cloud service, let S0 be the identity element of G1. The root PKG that is the cloud server S1 parent operates as follows:

Step 1: computes P1=H1(ID1) ∈ G1;

Step 2: set the cloud server S1 private key S1=s0P1;

Level-2 set up. Level-2 is the users. The user's private key is extracted in this phrase. Let 2-tuple (ID1,IDA) be an identity of Level-2 user A. The cloud server S1 that is the user's parent generates the private key as follows:

Step 1: computes PA = H1(ID1 || IDA) G1;

Step 2: sets the user's private key SA = S1 + s1PA;

Step 3: also gives the user the values of Q1 = s1P0 as "verification points "to the user and needs to return (SA, Q1).

Hierarchical Identity-based Authenticated Key Agreement (HIB-AKA)

The main problem here is that this digital identity can only be used in one cloud, private one or public one. Users in a hybrid cloud may want to access services that are provided by different clouds, so it needs multiple identities for each one of services on these clouds. Here it is shown clearly that it is not user friendly (Li *et al*., 2013; Han *et al*., 2013). In this paper, we propose a system for cloud computing environment where each user and server will have its own unique identity, with which the key distribution and mutual authentication can be greatly simplified.

This paper was proposed to solve this problem by using identity management in clouds computing with hierarchal identity based cryptography, where this proposed scheme allows users from one cloud to access service in another cloud with single digital identity, and also allows them in hybrid cloud to simplify a mutual authentication and key distribution. Our protocol design should achieve

the following security and performance guarantee, to enable privacy-preserving public accessing for cloud environment.

Alice and Bob wish to establish shared key. Each of them chooses a random element as private key a, b Z*q , and computes the values of corresponding element as public keys WA = aPA , WB = bPB and S1 = s0P1. They can exchange the public keys as following:

Alice sends a message M1 to Bob which contains WA. Bob sends a message M2 to Alice WB. Then they may produce the algorithm,where Alice computes:

$$KAB = e(SA, WB + aPB) / e(S1, WB + aPB)$$

$$= e(s0P1+s1PA, WB + aPB) / e(s0P1, WB + aPB)$$

$$= e(s1PA, WB + aPB)$$

$$= e(s1PA, bPB + aPB)$$

$$= e(PA, PB)^{s1(a+b)}$$

In addition, Bob computes:

$$KBA = e(WA + bPA, SB) / e(WA + bPA, S1)$$

$$= e(WA + bPA, s0P1 + s1PB) / e(WA + bPA, s0P1)$$

$$= e((a+b)PA, s1PB)$$

$$= e(PA,PB)^{s1(a+b)}$$

If Alice and Bob follow the algorithm, then they get the same share key.

Security Analysis of HIB-AKA Protocol

Our proposed algorithm is secure, because it has a hierarchical design. Regular identity-based cryptography has one PKG that distributes private keys to users. If the root PKG exposes the private key, all users' private keys are also revealed. In our proposed algorithm, level-2 users cannot influence if the root PKG reveals all private keys. Since they have different parents, the user's private keys are secure. In addition, any other domain that is not connected with the root PGK cannot expose their domain private keys. So it can greatly reduce the workload, also can allow key escrow at several levels.

If Alice assures that no other entity besides Bob can possibly ascertain the value of the secret key, then we can say that a key agreement protocol is implicit key authentication (Farash & Attari, 2014). When this condition is achieved,the mutual implicit key authentication is an authentication key agreement protocol (Nabil *et al*., 2013).

Here, we will analyse the security attributes of our proposed authenticated key agreement protocol HIB-AKA (Tsai & Lai, 2013).

Known-key Security: A unique secret session key should be created in each run of protocol. The compromise of one session key should not compromise other session keys.

Forward Secrecy: If long-term private keys of one or more entities are compromised, the secrecy of previously established session keys should not be affected. We say that a system has partial forward secrecy if some but not all of the entities (long-term keys) can be corrupted without compromising previously established session keys, and we say that a system has perfect forward secrecy if the long-term keys of all the entities involved may be corrupted without compromising any session key previously established by these entities. There is further perhaps a stronger notion of forward secrecy in identity-based systems, which is called PKG forward secrecy, which implies perfect forward secrecy. This is the idea that the PKG's long- term private key may be corrupted and hence all users' long-term private keys, without compromising the security of session keys, were previously established by any users.

Key-compromise Impersonation Resilience: Compromising an entity, Alice's long-term private key will allow an adversary to impersonate Alice, but it should not enable the adversary to impersonate other entities to Alice.

Unknown Key-share Resilience: An entity, Alice should not be able to be coerced into sharing a key with any entity adversary when in fact Alice thinks that she is sharing the key with another entity, Bob.

Key Control: Neither entity should be able to force the session key to be a preselected value.

Verification of HIB-AKA Protocol using Scyther Tool

Our proposed secure and efficient HIB-AKA algorithm for cloud computing has the following properties:

a. less computational cost, so to be more efficient.

b. hierarchy, due to cloud computing environment scalability and dynamic features.

c. one-round, less network overload, so more efficient.

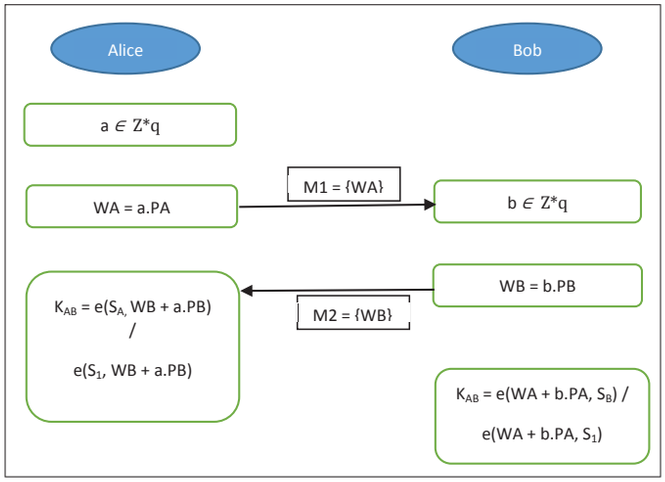
d. The users (i.e., Alice, Bob) can help to meet frequent mutual authentication requests between users and resources.

e. Unique node's registered distinguished name (DN) from root to node, to provide cross-trust domain in which each domain comprises one PKG.

Before authentication, trust relationship has built between PKGs to shared system parameters with each other.

If Alice and Bob follow the algorithm, then they get the same share key. Our proposed HIB-AKA algorithm is depicted in Figure 2.

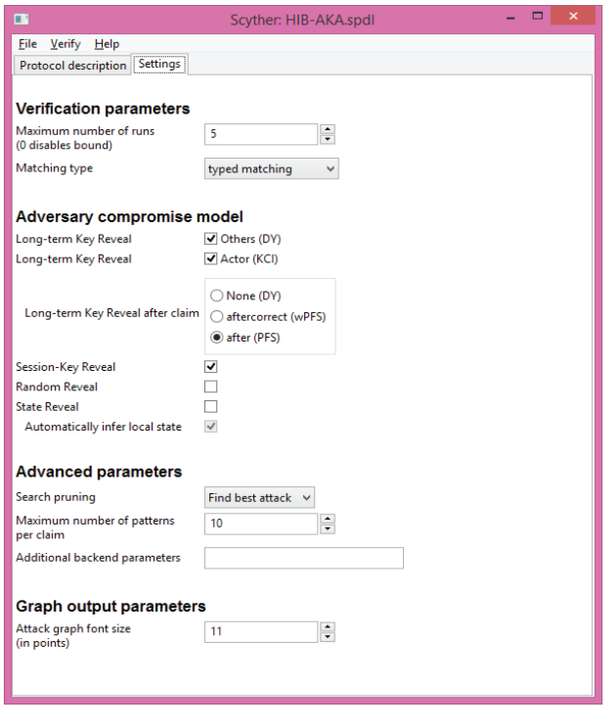
Figure 2: Proposed Key Agreement HIB-AKA



We use automated security protocol verification tool Scyther version compromise-0.9.2 to provide us a formal security analysis, on laptop 2.4 GHz Intel core i3 processor, with 3 GB RAM. Scyther tool presents a framework for modeling adversaries in security protocol analysis, ranging from a Dolev-Yao style adversary to more powerful adversaries, supports notions such as weak perfect forward secrecy, key compromise impersonation, and adversaries capable of state-reveal queries (Basin & Cremers, 2010).

Figure 3 shows the settings of the adversary model used in verifying our proposed HIB-AKA protocol.

Figure 3: Scyther Adversary Model Used for HIB-AKA Verifying



We model the HIB-AKA protocol in security protocol description language (SPDL) using Scyther tool as follows:

```
/* Proposed Hierarchal identity based authentication key agreement for Cloud (HIB-AKA) */

// Hash functions

hashfunction E; // E is e pairing function

// Addition, multiplication, division simply hashes

hashfunctionmult,add,div;

// The protocol description

protocol HIB-AKA(CS,A,B)

// CS = Cloud Server A = Alice as user, B = Bob as user

{

        const S1,PA,PB;

        role CS // Cloud Server

        {

                send_1(CS,A,S1);  // Publish public
```

```
params
        send_2(CS,B,S1);
    }

    role A // Alice as User
    {
        fresh a: Nonce; // Ephemeral Secret
        var WB: Ticket;

        recv_1(CS,A,S1);
        send_3(A,B,mult(a,PA)); // Send WA
        recv_4(B,A,WB);
        // Secret Session Key
        claim(A,SKR,div(E(sk(A,CS),add(WB,
mult(a,PB))), E(S1,add(WB,mult(a,PB)))));
    }

    role B // Bob as User
    {
        fresh b: Nonce; // Ephemeral Secret
        var WA: Ticket;
        recv_2(CS,B,S1);
        recv_3(A,B,WA);
        send_4(B,A,mult(b,PB)); // Send WB
        // Secret Session Key
        claim(B,SKR,div(E(add(WA,mult(b,PA)
),sk(B,CS)),E(add(WA,mult(b,PA),S1))));
    }
}
```

Figure 4 shows the proposed HIB-AKA verification using Scyther tool.

Conclusions

In this paper, we proposed an efficient identity-based authenticated key agreement protocol for configurable hierarchical cloud computing environment. There are

Figure 4: HIB-AKA Scyther Security Protocol Verification.

Scyther results : verify

Claim				Status	Comments
HIB_AKA	A	HIB_AKA,A1	SKR div(E(sk(A,CS),add(WB,mult(a,PB))),E(S1,add(WB...	Ok	No attacks within bounds.
	B	HIB_AKA,B1	SKR div(E(add(WA,mult(b,PA)),sk(B,CS)),E(add(WA,mu...	Ok	No attacks within bounds.

Done.

Scyther results : characterize

Claim				Status		Comments	Patterns
HIB_AKA	CS	HIB_AKA,CS1	Reachable	Ok	Verified	Exactly 1 trace pattern.	1 trace pattern
	A	HIB_AKA,A2	Reachable	Ok	Verified	Exactly 1 trace pattern.	1 trace pattern
	B	HIB_AKA,B2	Reachable	Ok	Verified	Exactly 1 trace pattern.	1 trace pattern

Done.

tools for specifying and verifying security protocols like AVISPA, CASRUL, ProVerif and Scyther tools. We had demonstrated a background of Hierarchical Identity-Based Key management as well as explained our proposed HIB-AKA security protocol through observing features and advantages of our proposed protocol, then we analysed security attributes of our proposed protocol. Finally our proposed protocol is provably secure using a formal security protocol verification Scyther Tool.

References

Alegría, J. A. H., Bastarrica, M. C., & Bergel, A. (2014). Avispa: A tool for analyzing software process models. *Journal of Software: Evolution and Process*, 26(4), 434-450.

Basin, D., & Cremers, C. (2010). Modeling and analyzing security in the presence of compromising adversaries. *In Computer Security-ESORICS 2010*(pp. 340-356). Berlin Heidelberg: Springer.

Basin, D., Cremers, C., & Meier, S. (2013). Provably repairing the ISO/IEC 9798 standard for entity authentication. *Journal of Computer Security*, 21(6), 817-846.

Basin, D., Cremers, C., Miyazaki, K., Radomirovic, S., & Watanabe, D. (2014). *Improving the Security of Cryptographic Protocol Standards*.

Blanchet, B. (2011). Using Horn clauses for analyzing security protocols.*Formal Models and Techniques for Analyzing Security Protocols*, 5, 86-111.

Boneh, D., & Franklin, M. (2003). Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3), 586-615.

Cao, X., Kou, W., & Du, X. (2010). A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges. *Information Sciences*, 180(15), 2895-2903.

Chatterjee, S., & Sarkar, P. (2011). *Identity-based encryption*. Springer.

Cheval, V., & Blanchet, B. (2013). Proving more observational equivalences with ProVerif. *In Principles of Security and Trust* (pp. 226-246). Springer Berlin Heidelberg.

Cortier, V., & Warinschi, B. (2005). Computationally sound, automated proofs for security protocols. *In Programming Languages and Systems* (pp. 157-171). Springer Berlin Heidelberg.

Cremers, C., & Mauw, S. (2012). *Operational semantics and verification of security protocols* (pp. 1-155). Berlin: Springer.

Elbaz, H. A., Abd-elaziz, M. H., & Nazmy, T. (2014). Trusting identity based authentication on hybrid cloud computing. *In Cloud Computing 2010*(pp. 179-188). Springer International Publishing.

Farash, M. S., & Attari, M. A. (2014). An Enhanced and Secure Three-Party Password-based Authenticated Key Exchange Protocol without Using Server's Public-Keys and Symmetric Cryptosystems. *Information Technology and Control*, 43(2), 143-150.

Farash, M. S., Attari, M. A., Atani, R. E., & Jami, M. (2013). A new efficient authenticated multiple-key exchange protocol from bilinear pairings. *Computers & Electrical Engineering*, 39(2), 530-541.

Han, J., Susilo, W., & Mu, Y. (2013). *Identity-based secure distributed data storage schemes*.

Li, Y., Du, L., Zhao, G., & Guo, J. (2013, August). *A Lightweight Identity-based Authentication Protocol*. IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC)(pp. 1-4).

Luu, A. T., Sun, J., Liu, Y., Dong, J. S., Li, X., & Quan, T. T. (2012). SeVe: Automatic tool for verification of security protocols. *Frontiers of Computer Science*, 6(1), 57-75.

Martin, L. (2008). *Introduction to identity-based encryption*. Artech house.

Modersheim, S. (2009). *Algebraic Properties in Alice and Bob Notation. In Availability, Reliability and Security, 2009*. International Conference on ARES 09. (pp. 433-440).

Nabil, M., Abouelseoud, Y., Elkobrosy, G., & Abdelrazek, A. (2013). *New Authenticated Key Agreement Protocols*. In Proceedings of the International MultiConference of Engineers and Computer Scientists.

Tsai, C. C.,& Lai, P. J. (2013). Analysis of authenticated key agreement protocols from weil paring. *SCA: International Journal of Soft Computing with Applications*, 1(1), 10-19.

Yao, A. C., & Zhao, Y. (2014). *Privacy-Preserving Authenticated Key-Exchange Over Internet*.