ID2S Password Authenticated Key Exchange Protocol

A. KALAVATHI [1], VARSHITHA M.[2], M. SAHITHYA[3], T. SRAVYA[4]
[1,2,3,4] Dept. of Information Technology, VVIT, AP, India

Abstract -- In two-server password-authenticated key exchange (PAKE) protocol, a client splits its password and stores two shares of its password in the two servers, respectively, and the two servers then cooperate to authenticate the client without knowing the password of the client. In case one server is compromised by an adversary, the password of the client is required to remain secure. In this paper, we present two compilers that transform any two-party PAKE protocol to a two-server PAKE protocol on the basis of the identity-based cryptography, called ID2S PAKE protocol. By the compilers, we can construct ID2S PAKE protocols which achieve implicit authentication. As long as the underlying two-party PAKE protocol and identity-based encryption or signature scheme have provable security without random oracles, the ID2S PAKE protocols constructed by the compilers can be proven to be secure without random oracles. Compared with the Katz et al.'s two-server PAKE protocol with provable security without random oracles, our ID2S PAKE protocol can save from 22% to66%.

I. INTRODUCTION

To secure communication between two parties, an authenticated encryption key is required to agree on in advance. So far, two models have existed for authenticated key exchange. One model assumes that two parties already share some cryptographically-strong information: either a secret key which can be used for encryption/authentication of messages, or a public key which can be used for encryption/ signing of messages. These keys are random and hard to remember. In practice, a user often keeps his keys in a personal device protected by a password/PIN. Another model assumes that users, without help of personal devices, are only capable of storing "human-memorable" passwords. Bellovin and Merritt [4] were the first to introduce password-based authenticated key exchange (PAKE), where two parties, based only on their knowledge of a password, establish a cryptographic key by exchange of messages. A PAKE protocol has to be immune to on-line and off-line dictionary attacks. In an off-line dictionary attack, an adversary exhaustively tries all possible passwords in a dictionary in order to determine the password of the client on the basis of the exchanged messages. In on-line dictionary attack, an adversary simply attempts to login repeatedly, trying each possible password. By cryptographic means only, none of PAKE protocols can prevent on-line dictionary attacks. But on-line attacks can be stopped simply by setting a threshold.

1) Password-only PAKE: Typical examples are the "encrypted key exchange" (EKE) protocols given by Bellovin and Merritt, where two parties, who share a password, exchange messages encrypted by the password, and establish a common secret key. The formal model of security for PAKE was firstly given in. Based on the security model, PAKE protocols have been proposed and proved to be secure.

2) PKI-based PAKE: PKI-based PAKE protocol was first given by Gong et al., where the client stores the server's public key in addition to share a password with the server. Halevi and Krawczyk were the first to provide formal definitions and rigorous proofs of security for PKI-based PAKE.

3) ID-based PAKE: ID-based PAKE protocols were proposed by Yi et al., where the client needs to remember a password in addition to the identity of the server, whereas the server keeps the password in addition to a private key related to its identity. ID-based PAKE can be thought as a trade-off between password-only and PKI-based PAKE.In the single-server setting, all the passwords necessary to authenticate clients are stored in a single server. If the server is compromised, due to, for example, hacking or even insider attacks, passwords stored in the server are all disclosed. This is also true to Kerberos, where a user authenticates against the authentication server with his username and password and obtains a token to authenticate against the service server.

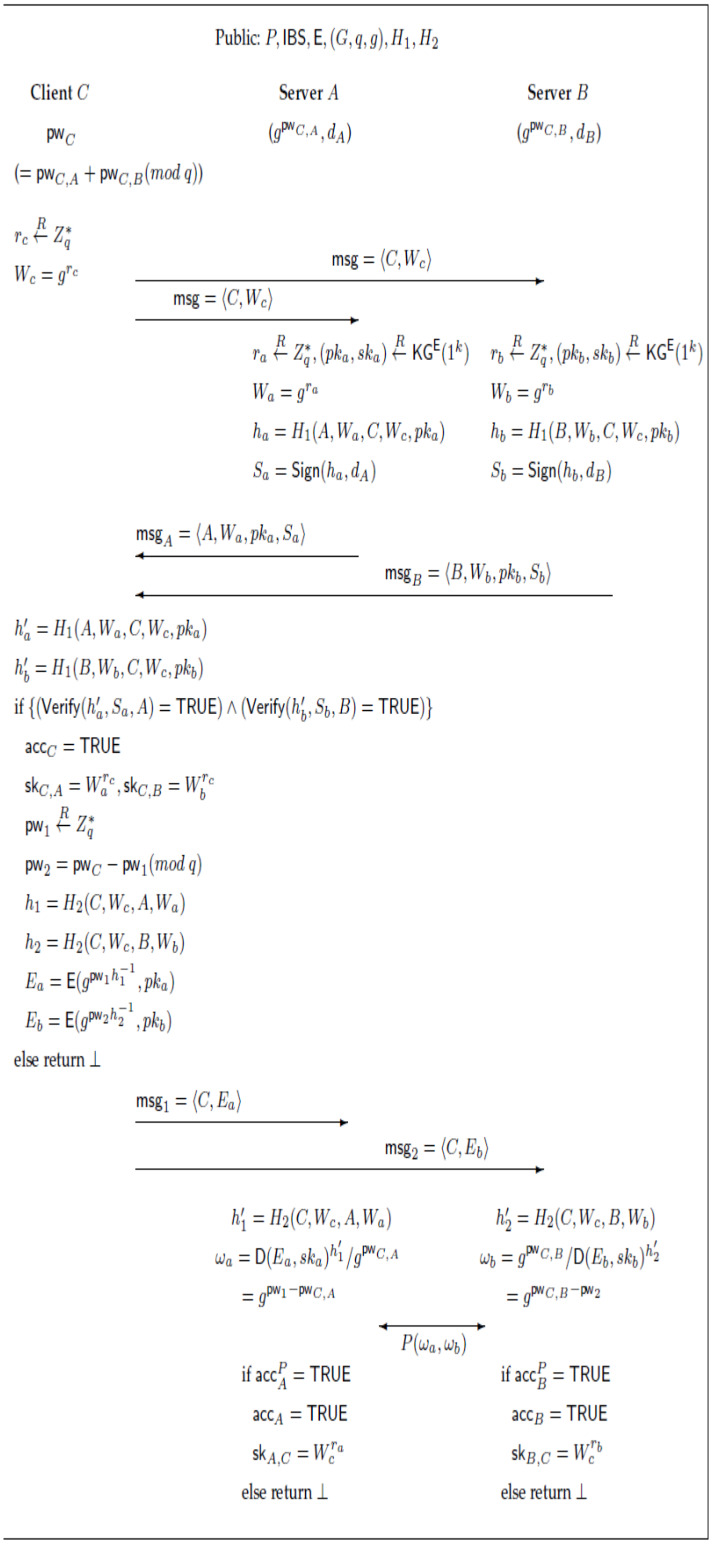
II. MODULES DESCRIPTION

We present two compilers transforming any two-party PAKE protocol P to an ID2S PAKE protocol P0 with identity-based cryptography. The first compiler is built on identity-based signature (IBS) and the second compiler is based on identity-based encryption (IBE).

1) ID2S PAKE Based on IBS: We need an identity-based signature scheme (IBS) as our cryptographic building block. A high-level description of our compiler in which the client C and two servers A and B establish two authenticated keys, respectively. If we remove authentication elements from our compiler, our key exchange protocol is essentially the Diffie-Hellman key exchange protocol. We present the protocol by describing initialization and execution.

The Diffie-Hellman key exchange protocol was invented by Diffie and Hellman in 1976. It was the first practical method for two users to establish a shared secret key over an unprotected communications channel. Although it is a non-authenticated key exchange protocol, it provides the basis for a variety of authenticated protocols. Diffie-Hellman key exchange protocol was followed shortly afterward by RSA, the first practical public key cryptosystem.

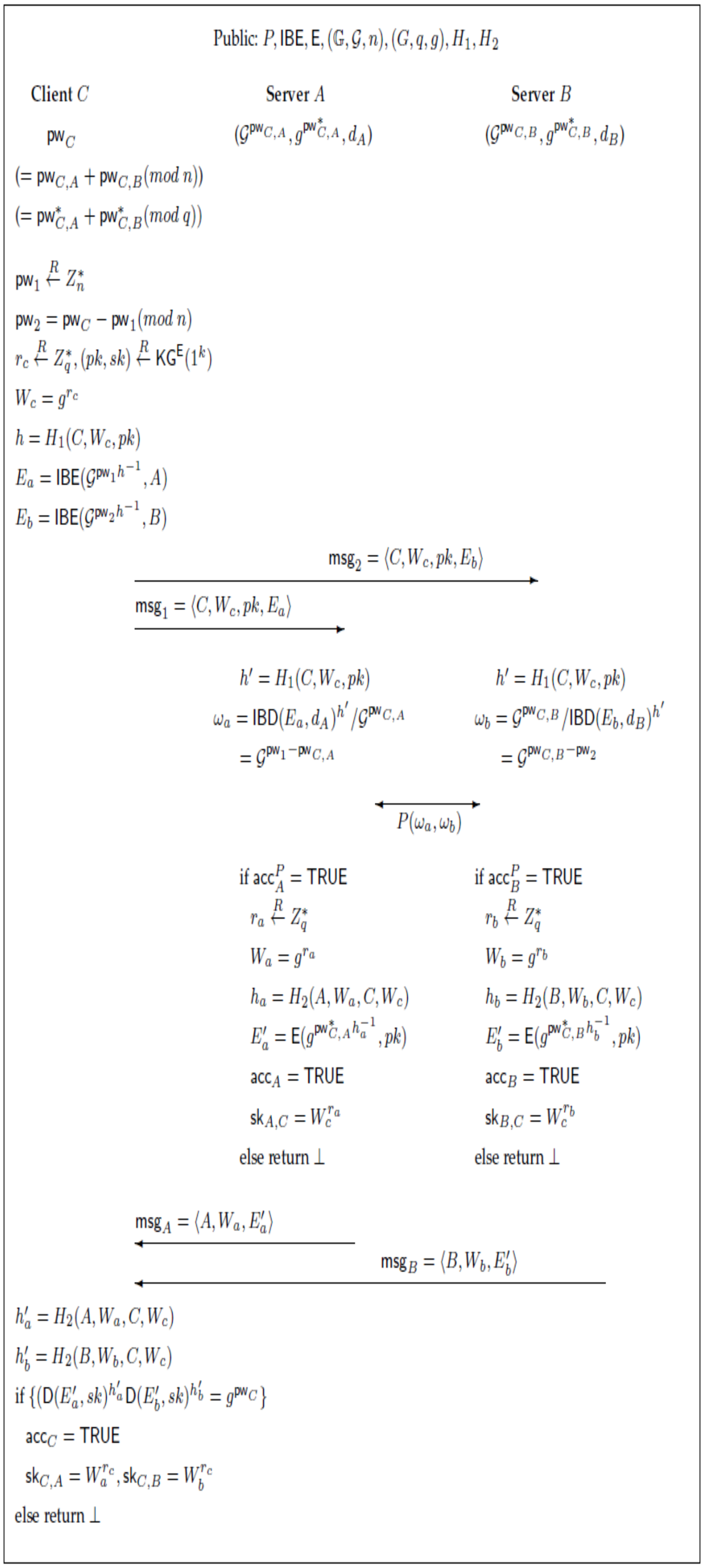
Key Generation: On input the identity S of a server S 2 Server, paramsIBS, and the secret sharing master-keyIBS, PKGs cooperate to run ExtractIBS of the IBS scheme and generate a private (signing) key for S, denoted as dS, in a manner that any coalition of PKGs cannot determine dS as long as one of the PKGs is honest to follow the protocol.


Public: P, IBS, E, $(G,q,g), H_1, H_2$
Client C
Server A
Server B
pw_C
$(g^{\mathsf{pw}_{C,A}}, d_A)$
$(g^{\mathsf{pw}_{C,B}}, d_B)$
$(= \mathsf{pw}_{C,A} + \mathsf{pw}_{C,B} (mod\ q))$
$r_c \overset{R}{\leftarrow} Z_q^*$
$W_c = g^{r_c}$
$\mathsf{msg} = \langle C, W_c \rangle$
$\mathsf{msg} = \langle C, W_c \rangle$
$r_a \overset{R}{\leftarrow} Z_q^*, (pk_a, sk_a) \overset{R}{\leftarrow} \mathsf{KG}^{\mathsf{E}}(1^k)$
$W_a = g^{r_a}$
$h_a = H_1(A, W_a, C, W_c, pk_a)$
$S_a = \mathsf{Sign}(h_a, d_A)$
$r_b \overset{R}{\leftarrow} Z_q^*, (pk_b, sk_b) \overset{R}{\leftarrow} \mathsf{KG}^{\mathsf{E}}(1^k)$
$W_b = g^{r_b}$
$h_b = H_1(B, W_b, C, W_c, pk_b)$
$S_b = \mathsf{Sign}(h_b, d_B)$
$\mathsf{msg}_A = \langle A, W_a, pk_a, S_a \rangle$
$\mathsf{msg}_B = \langle B, W_b, pk_b, S_b \rangle$
$h'_a = H_1(A, W_a, C, W_c, pk_a)$
$h'_b = H_1(B, W_b, C, W_c, pk_b)$
if $\{(\mathsf{Verify}(h'_a, S_a, A) = \mathsf{TRUE}) \wedge (\mathsf{Verify}(h'_b, S_b, B) = \mathsf{TRUE})\}$
$\mathsf{acc}_C = \mathsf{TRUE}$
$\mathsf{sk}_{C,A} = W_a^{r_c}, \mathsf{sk}_{C,B} = W_b^{r_c}$
$\mathsf{pw}_1 \overset{R}{\leftarrow} Z_q^*$
$\mathsf{pw}_2 = \mathsf{pw}_C - \mathsf{pw}_1 (mod\ q)$
$h_1 = H_2(C, W_c, A, W_a)$
$h_2 = H_2(C, W_c, B, W_b)$
$E_a = \mathsf{E}(g^{\mathsf{pw}_1 h_1^{-1}}, pk_a)$
$E_b = \mathsf{E}(g^{\mathsf{pw}_2 h_2^{-1}}, pk_b)$
else return $\perp$
$\mathsf{msg}_1 = \langle C, E_a \rangle$
$\mathsf{msg}_2 = \langle C, E_b \rangle$
$h'_1 = H_2(C, W_c, A, W_a)$
$\omega_a = \mathsf{D}(E_a, sk_a)^{h'_1} / g^{\mathsf{pw}_{C,A}}$
$= g^{\mathsf{pw}_1 - \mathsf{pw}_{C,A}}$
$h'_2 = H_2(C, W_c, B, W_b)$
$\omega_b = g^{\mathsf{pw}_{C,B}} / \mathsf{D}(E_b, sk_b)^{h'_2}$
$= g^{\mathsf{pw}_{C,B} - \mathsf{pw}_2}$
$P(\omega_a, \omega_b)$
if $\mathsf{acc}_A^P = \mathsf{TRUE}$
$\mathsf{acc}_A = \mathsf{TRUE}$
$\mathsf{sk}_{A,C} = W_c^{r_a}$
else return $\perp$
if $\mathsf{acc}_B^P = \mathsf{TRUE}$
$\mathsf{acc}_B = \mathsf{TRUE}$
$\mathsf{sk}_{B,C} = W_c^{r_b}$
else return $\perp$

2) ID2S PAKE Based on IBE: A high-level description of our compiler based on identitybased encryption. We present the protocol by describing initialization and execution.

Key Generation: On input the identity S of a server S 2 Server, paramsIBE, and the secret sharing master-keyIBE, PKGs cooperate to run ExtractIBE of the IBE scheme and generate a private (decryption) key for S, denoted as dS, in a manner that any coalition of PKGs cannot determine dS as long as one of the PKGs is honest to follow the protocol.
Each user has a private key x
Each user has three public keys: prime modulus p, generator g and public Y = gxmod p
Security is based on the difficulty of DLP
Secure key size > 1024 bits (today even 2048 bits)
Elgamal is quite slow, it is used mainly for key authentication protocols
Protocol Execution. Given a triple (C; A;B) 2 Client ServerTriple, the client C (knowing its password pwC) runs the protocol P0 with the two servers A (knowing GpwC;A , gpwC;A and its private key dA) and B (knowing GpwC;B , gpw C;B and its private key dB) to establish two session keys, respectively.

At first, the client randomly chooses pw1 from Zn and computes pw2 = pwC □ pw1(mod n). Next the client C randomly generates a one-time public and private key pair (pk; sk) for the public key encryption scheme E, and randomly chooses an integer rc from Zq and computes Wc = grc ; h = H1(C;Wc; pk): Next, according to the identities of the two servers A and B, the client C performs the identity-based encryptions Ea = IBE(Gpw1h□1 ;A);Eb = IBE(Gpw2h□1 ;B): Then, the client sends msg1 = hC;Wc; pk;Eai and msg2 = hC;Wc; pk;Ebi to the two servers A and B, respectively.


Public: $P, \mathsf{IBE}, \mathsf{E}, (\mathbb{G}, \mathcal{G}, n), (G, q, g), H_1, H_2$
Client C
Server A
Server B
pw_C
$(\mathcal{G}^{\mathsf{pw}_{C,A}}, g^{\mathsf{pw}^*_{C,A}}, d_A)$
$(\mathcal{G}^{\mathsf{pw}_{C,B}}, g^{\mathsf{pw}^*_{C,B}}, d_B)$
$(= \mathsf{pw}_{C,A} + \mathsf{pw}_{C,B} (mod\ n))$
$(= \mathsf{pw}^*_{C,A} + \mathsf{pw}^*_{C,B} (mod\ q))$
$\mathsf{pw}_1 \overset{R}{\leftarrow} Z_n^*$
$\mathsf{pw}_2 = \mathsf{pw}_C - \mathsf{pw}_1 (mod\ n)$
$r_c \overset{R}{\leftarrow} Z_q^*, (pk, sk) \overset{R}{\leftarrow} \mathsf{KG}^{\mathsf{E}}(1^k)$
$W_c = g^{r_c}$
$h = H_1(C, W_c, pk)$
$E_a = \mathsf{IBE}(\mathcal{G}^{\mathsf{pw}_1 h^{-1}}, A)$
$E_b = \mathsf{IBE}(\mathcal{G}^{\mathsf{pw}_2 h^{-1}}, B)$
$\mathsf{msg}_2 = \langle C, W_c, pk, E_b \rangle$
$\mathsf{msg}_1 = \langle C, W_c, pk, E_a \rangle$
$h' = H_1(C, W_c, pk)$
$\omega_a = \mathsf{IBD}(E_a, d_A)^{h'} / \mathcal{G}^{\mathsf{pw}_{C,A}}$
$= \mathcal{G}^{\mathsf{pw}_1 - \mathsf{pw}_{C,A}}$
$h' = H_1(C, W_c, pk)$
$\omega_b = \mathcal{G}^{\mathsf{pw}_{C,B}} / \mathsf{IBD}(E_b, d_B)^{h'}$
$= \mathcal{G}^{\mathsf{pw}_{C,B} - \mathsf{pw}_2}$
$P(\omega_a, \omega_b)$
if $\mathsf{acc}_A^P = \mathsf{TRUE}$
$r_a \overset{R}{\leftarrow} Z_q^*$
$W_a = g^{r_a}$
$h_a = H_2(A, W_a, C, W_c)$
$E'_a = \mathsf{E}(g^{\mathsf{pw}^*_{C,A} h_a^{-1}}, pk)$
$\mathsf{acc}_A = \mathsf{TRUE}$
$\mathsf{sk}_{A,C} = W_c^{r_a}$
else return $\perp$
if $\mathsf{acc}_B^P = \mathsf{TRUE}$
$r_b \overset{R}{\leftarrow} Z_q^*$
$W_b = g^{r_b}$
$h_b = H_2(B, W_b, C, W_c)$
$E'_b = \mathsf{E}(g^{\mathsf{pw}^*_{C,B} h_b^{-1}}, pk)$
$\mathsf{acc}_B = \mathsf{TRUE}$
$\mathsf{sk}_{B,C} = W_c^{r_b}$
else return $\perp$
$\mathsf{msg}_A = \langle A, W_a, E'_a \rangle$
$\mathsf{msg}_B = \langle B, W_b, E'_b \rangle$
$h'_a = H_2(A, W_a, C, W_c)$
$h'_b = H_2(B, W_b, C, W_c)$
if $\{(\mathsf{D}(E'_a, sk)^{h'_a} \mathsf{D}(E'_b, sk)^{h'_b} = g^{\mathsf{pw}_C}\}$
$\mathsf{acc}_C = \mathsf{TRUE}$
$\mathsf{sk}_{C,A} = W_a^{r_c}, \mathsf{sk}_{C,B} = W_b^{r_c}$
else return $\perp$

III. INITIALIZATION

The two peer servers S1 and S2 jointly choose a cyclic group G of large prime order q with a generator g1 and a secure hash function H : {0; 1}*->Zq, which maps a message of arbitrary length into an l-bit integer, where l= log2 q. Next, S1 randomly chooses an integer s1 from Zq and S2 randomly chooses an integer s2 from Zq , and S1 and S2 exchange g1s1 and g1s2 . After that, S1 and S2 jointly publish public system parameters G; q; g1; g2;H where g2 = gs1s2.

IV. REGISTRATION

The two secure channels are necessary for all two server PAKE protocols, where a password is split into two parts, which are securely distributed to the two servers, respectively, during registration. Although we refer to the concept of public key cryptosystem, the encryption key of one server should be unknown to another server and the client needs to remember a password only after registration.

V. CONCLUSION

This system work presents two efficient compilers to transform any two-party PAKE protocol to an ID2S PAKE protocol with identity-based cryptography. In addition, we have provided a rigorous proof of security for our compilers without random oracle. Our compilers are in particular suitable for the applications of password-based authentication where an identity-based system has already established. Our future work is to construct an identity-based multiple server PAKE protocol with any two-party PAKE protocol.

VI. FUTURE WORK

For the purpose of helping the data owner enjoy ne-grained access control of data stored on un trusted servers, a feasible solution would be encrypting data through certain cryptographic primitive(s), and disclosing decryption. Moreover, the new cryptographic primitive(s) needs to be able to support dynamic requests so that data owners can add or revoke access privileges to other users. Therefore, one critical issue with this branch of approaches is how to achieve the desired security goals outlined above without introducing high complexity on computation, privilege revocation and key management.

This can be further worked by keeping the OTP checks on the given mobile number.

REFERENCES

[1] X. Yi, E. Bertino, J. Vaidya, and C. Xing, “Private searching on streaming data based on keyword frequency,” IEEE Transactions on Dependable and Secure Computing, vol. 11, no. 2, pp. 155–167, 2014.

[2] P. Xu, H. Jin, Q. Wu, and W. Wang, “Public-key encryption with fuzzy keyword search: A provably secure scheme under keywordguessing attack,” IEEE Trans. Computers, vol. 62, no. 11, pp. 2266– 2277, 2013.

[3] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, “Dual-server public-key encryption with keyword search for secure cloud storage,” IEEE Transactions on Information Forensics and Security, vol. 11, no. 4, pp. 789–798, 2016.

[4] P. Xu, Q. Wu, W. Wang, W. Susilo, J. Domingo-Ferrer, and H. Jin, “Generating searchable public-key ciphertexts with hidden structures for fast keyword search,” IEEE Transactions on Information Forensics and Security, vol. 10, no. 9, pp. 1993–2006, 2015.

[5] H. S. Rhee, J. H. Park,W. Susilo, and D. H. Lee, “Trapdoor security in a searchable public-key encryption scheme with a designated tester,” Journal of Systems and Software, vol. 83, no. 5, pp. 763–771, 2010.

[6] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” in ACM CCS, 2006, pp. 79–88.

[7] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in EUROCRYPT, 2004, pp. 506–522.

[8] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, “A new general framework for secure public key encryption with keyword search,” in ACISP, 2015, pp. 59–76.

[9] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions," in CRYPTO, 2005, pp. 205–222.

[10] D. Khader, "Public key encryption with keyword search based on k-resilient IBE," in Computational Science and Its Applications - ICCSA, 2006, pp. 298–308.