"IMPLEMENTATION OF IDENTITY-BASED CRYPTOGRAPHY FOR ALLOCATING RESOURCES IN CLOUD COMPUTING"

[1]MR JAICKY R. SANCHETI
Computer Science and Engineering, H.V.P.M.'s College of Engineering & Technology, Amravati, India
jaickyrsancheti@gmail.com

[2]DR. A. B. RAUT
Computer Science and Engineering, H.V.P.M.'s College of Engineering & Technology, Amravati, India
anjali_dahake@rediffmail.com

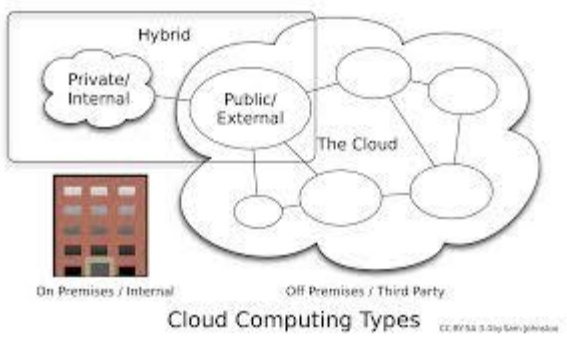
ABSTRACT: *More and more companies begin to provide different kinds of cloud computing services for Internet users at the same time these services also bring some security problems. Currently the majority of cloud computing systems provide digital identity for users to access their services, this will bring some inconvenience for a hybrid cloud that includes multiple private clouds and/or public clouds. Today most cloud computing system use asymmetric and traditional public key cryptography to provide data security and mutual authentication. Identity-based cryptography has some attraction characteristics that seem to fit well the requirements of cloud computing. In this paper, by adopting federated identity management together with hierarchical identity-based cryptography (HIBC), not only the key distribution but also the mutual authentication can be simplified in the cloud..*

Keywords: Cloud Computing Security, Cloud Computing Identity Management

1. INTRODUCTION

Cloud Computing is a new computing model that distributes the computing missions on a resource pool that includes a large amount of computing resources. It is the result of development of infrastructure as a service (IAAS), platform as a service (PAAS), and software as a service (SAAS). With broadband Internet access, Internet users are able to acquire computing resource, storage space and other kinds of software services according to their needs. In cloud computing, with a large amount of various computing resources, users can easily solve their problems with the resources provided by a cloud. This brings great flexibility for the users. Using cloud computing service, users can store their critical data in servers and can access their data anywhere they can with the Internet and do not need to worry about system breakdown or disk faults, etc. Also, different users in one system can share their information and work, as well as play games together. Many important companies such as Amazon, Google, IBM, Microsoft, and Yahoo are the forerunners that provide cloud computing services. Recently more and more companies such as Sales force, Facebook, Youtube, Myspace etc. also begin to provide all kinds of cloud computing services for Internet users.

Currently, as shown in Figure 1, there are mainly three types of clouds: private clouds, public clouds and hybrid clouds [15]. Private clouds, also called internal clouds, are the private networks that offer cloud computing services for a very restrictive set of users within internal network. For example, some companies and universities can use their internal networks to provide cloud computing services for their own users. These kinds of networks can be thought as private clouds. Public clouds or external clouds refer to clouds in the traditional sense [13], such as enterprises that provide cloud computing services for the public users. Hybrid clouds are the clouds that include multiple private and/or public clouds [14]. Providing security in a private cloud and a public cloud is easier, comparing with a hybrid cloud since commonly a private cloud or a public cloud only has one service provider in the cloud. Providing security in a hybrid cloud that consisting multiple service providers is much more difficult especially for key distribution and mutual authentication. Also for users to access the services in a cloud, a user digital identity is needed for the servers of the cloud to manage the access control. While in the whole cloud, there are many different kinds of clouds and each of them has its own identity management system. Thus user who wants to access services from different clouds needs multiple digital identities from different clouds, which will bring inconvenience for users. Using federated identity management, each user will have his unique digital identity and with this identity, he can access different services from different clouds. Identity-based cryptography [10] is a public key technology that allows the use of a public identifier of a user as the user's public key. Hierarchy identity-based cryptography is the development from it in order to solve the scalability problem. Recently identity-based cryptography and hierarchy identity-based cryptography have been proposed to provide security for some Internet applications. For example, applying identity-based cryptography in the grid computing and web service security have been explored in [11] [8] [12] and [5]. This paper proposes to use federated identity management in the cloud such that each user and each server will have its own unique identity, and the identity is allocated by the system hierarchically. With this unique identity and hierarchical identity based cryptography (HIBC), the key distribution and mutual authentication can be greatly simplified.



The rest of this paper is organized as follows. In Section 2, we introduce security problems and related solutions in cloud computing. In Section 3, we describe the principle of identity-based cryptography and HIBC. In Section 4, we describe how to use federated identity management and HIBC in the cloud computing system to provide security. Section 5 concludes the paper.

2. SECURITY IN CLOUD COMPUTING CLOUD

Computing has many advantages in cost reduction, resource sharing, and time saving for new service deployment. While in a cloud computing system, most data and software that users use reside on the Internet, which bring some new challenges for the system, especially security and privacy. Since each application may use resource from multiple servers. The servers are potentially based at multiple locations and the services provided by the cloud may use different infrastructures across organizations. All these characteristics of cloud computing make it complicated to provide security in cloud computing. To ensure adequate security in cloud computing, various security issues, such as authentication, data confidentiality and integrity, and non-repudiation, all need to be taken into account. Currently, WS-Security service is wildly used in the cloud to provide security for the system. In WS-Security, XML encryption and XML signature are used to provide data confidentiality and integrity. Mutual authentication can be supported by adding X.509 certificate and Kerberos tickets into SOAP message header. As mentioned earlier, there are three types of clouds in general: private cloud, public cloud and hybrid cloud. In a public cloud, resources are dynamically provisioned on a fine-grained, self-service basis over the Internet. Services in the cloud are provided by an off-site third-party provider who shares resources and bills on a fine-grained utility computing basis. While in most private clouds, with limited computing resources, it is difficult for a private cloud to provide all services for their users, as some services may more resources than internal cloud can provide. Hybrid cloud is a potential solution for this issue since they can get the computing resources from external cloud computing providers. Private clouds have their advantages in corporation governance and offer reliable services, as well as they allow more control than public clouds do. For the security concerns, when a cloud environment is created inside a firewall, it can provide its users with less exposure to Internet security risks. Also in the private cloud, all the services can be accessed through internal connections rather than public Internet connections, which make it easier to use existing security measures and standards. This can make private clouds more appropriate for services with sensitive data that must be protected. While in a hybrid cloud, it includes more than one domain, which will increase the difficulty of security provision, especially key management and mutual authentication. The domains in a hybrid cloud can be heterogeneous networks; hence there may be gaps between these networks and between the different services providers. Even security can be well guaranteed in each of private/public cloud, while in a hybrid cloud with more than one kind of clouds that have different kinds of network conditions and different security policies, how to provide efficient security protection is much more difficult. For example, cross domain authentication can be a problem in a hybrid cloud with different domains. Although some authentication services such as Kerberos can provide multi-domain authentication, but one of the requirements for the multi-domain Kerberos authentication is that the Kerberos server in each domain needs to share a secret key with servers in other Kerberos domains and every two Kerberos servers need to be registered with each other. The problem here is if there are N Kerberos domains and each of them wants to trust each other, then the number of key exchanges is N (N-1)/2. For a hybrid cloud with a large number of domains, this will bring a problem for scalability. If different networks in a hybrid cloud using different authentication protocols, this problem can be more complex. In a cloud, the cloud computing system needs to provide a strong and user-friendly way for users to access all kinds of services in the system. When a user wants to run an application in the cloud, the user is required to provide a digital identity. Normally, this identity is a set of bytes that related to the user. Based on the digital identity, a cloud system can know what right this user has and what the user is allowed to do in the system. Most of cloud platforms include an identity service since identity information is required for most distributed applications [3]. These cloud computing systems will provide a digital identity for every user. For example, user with a Windows Live ID can use cloud computing services provided by Microsoft and user who wants to access cloud computing services from Amazon and Google also needs an Amazon defined identity and Google account. Here, each of these companies is a public cloud. The problem here is this digital identity can only be used in one private cloud or one public cloud. Users want to access services in the cloud that provided by different clouds will need to have multiple identities, each for one of the cloud. This is obviously not user friendly. To solve these problems in the cloud, we propose to use federated identity management in clouds with HIBC. The proposed scheme does not only allow users from a cloud to access services from other clouds with a single digital identity, it also simplifies the key distribution and mutual authentication in a hybrid cloud.

3. Identity-Based Cryptography and Signature

Identity-based cryptography and signature schemes were firstly proposed by Shamir [10] in 1984. But only in 2001, an efficient approach of identity-based encryption schemes was developed by Dan Boneh and Matthew K. Franklin [2] and Clifford Cocks [4]. These schemes are based on bilinear pairings on elliptic curves and have provable security. Recently hierarchical identity-based cryptography (HIBC) has been proposed in [6, 7] to improve the scalability of traditional identity-based cryptography scheme. Identity-based cryptographic scheme is a kind of public-key based approach that can be used for two parties to exchange messages and effectively verify each other's signatures. Unlike in traditional public-key systems that using a random string as the public key, with identity-based cryptography users identity that can uniquely identify that user is used as the public key for encryption and signature verification. Identity based cryptography can ease the key management complexity as public keys are not required to be distributed securely to others. Another advantage of identity-based encryption is that encryption and decryption can be conducted offline without the key generation center.

In the identity-based cryptography approach, the PKG should creates a "master" public key and a corresponding "master" private key firstly, then it will make this "master" public key public for all the interested users. Any user can use this "master" public key and the identity of a user to create the public key of this user. Each user wants to get his private key needs to contact the PKG with his identity. PKG will use the identity and the "master" private key to generate the private key for this user. In Dan Boneh and Matthew K. Franklin's approach, they defined four algorithms for a complete identity-based cryptography system. It includes setup, extract, encryption and decryption. 1. Setup: PKG create a master key Km and the system parameters P. Km is kept secret and used to generate private key for users. System parameters P are made public for all the users and can be used to generate users' public key with their identities. 2. Extract: When a user requests his private key from the PKG, PKG will use the identity of this user, system parameters P and master key Km to generate a private key for this user. 3. Encryption: When a user wants to encrypt a message and send to another user, he can use the system parameters P, receiver's identity and the message as input to generate the cipher text. 4. Decryption: Receiving a cipher text, receiver can use the system parameters P and his private key got from the PKG to decrypt the cipher text. In a network using identity-based cryptography, the PKG needs not only to generate private keys for all the users, but also to verify the user identities and establish secure channels to transmit private keys. In a large network with only one PKG, the PKG will have a burdensome job. In this case, HIBC [6] can be a better choice. In a HIBC network, a root PKG will generate and distribute private keys for domain-level PKGs and the domain-level PKGs will generate and distribute private keys to the users in their own domain. HIBC is suitable for a large scale network since it can reduce the workload of root PKG by distribute the work of user authentication, private key generation and distribution to the different level of PKGs. It can also improve the security of the network because user authentication and private key distribution can be done locally. The HIBC encryption and signature algorithms include root setup, lower-level setup, extraction, encryption, and decryption. 1. Root setup: root PKG will generate the root PKG system parameters and a root secret. The root secret will be used for private key generation for the lower-level PKGs. The root system parameters are made publicly available and will be used to generate public keys for lower-level PKGs and users. 2. Lower-level setup: Each lower-level PKG will get the root system parameters and generate its own lower-level secret. This lower-level secret will be used to generate private keys for the users in its domain. 3. Extract: When a user or PKG at level t with its identity (1,..., t ID ID) requests his private key from its upper-level PKG, where (1,..., i ID ID) is the identity of its ancestor at level i ($1 \leq i \leq t$), the upper-level PKG will use this identity, system parameters and its own private key to generate a private key for this user. 4. Encryption: User who wants to encrypt a message M can use the system parameters, receiver's identity and the message as input to generate the cipher text. C = Encryption (parameters, receiver ID, M). 5. Decryption: Receiving a cipher text, receiver can use system parameters and his private key got from the PKG to decrypt the cipher text. M = Decryption (parameters, k, C), k is the private key of the receiver 6. Signing and verification: A user can use parameters, its private key, and message M to generate a digital signature and sends to the receiver. Receiver and verify the signature using the parameters, message M, and the sender's ID. Signature = Signing (parameters, k, M), k is the sender's private key. Verification = (parameters, sender ID, M, Signature). There are some inherent limitations with the identity-based cryptography [1]. One of the issues is the key escrow problem. Since users' private keys are generated by PKG, the PKG can decrypt a user's message and create any user's digital signature without authorization. This in fact means that PKGs must be highly trusted. So the identity based scheme is more appropriate for a closed group of users such as a big company or a university. Since only under this situation, PKGs can be set up with users' trust. In a system using HIBC, every PKG in the hierarchy knows the users' private keys in the domain under the PKG. Although key escrow problem cannot be avoided, this can limit the scope of key escrow problem. Another drawback of the identity-based cryptography is the revocation problem. Because all the users in the system use some unique identifiers as their public keys, if one user's private key has been compromised, the user need to change its public key. For example, if the public key is the user's name, address, or email address, it is inconvenient for the user to change it. One solution for this problem is to add a time period to the identifier as the public key [2], but it cannot solve this problem completely.

4. USING FEDERATED IDENTITY MANAGEMENT IN CLOUD

4.1 Federated Identity Management in the Cloud

Compared with centralized identity, which is used to deal with security problems within the same networks, federated identity is adopted to deal with the security problems that a user may want to access external networks or an external user may want to access internal networks. Federated identity is a standard-based mechanism for different organization to share identity between them and it can enable the portability of identity information to across different networks. One common use of federated identity is secure Internet single sign-on, where a user who logs in successfully at one organization can access all partner networks without having to log in again. Using identity federation can increase the security of network since it only requires a user to identify and authenticate him to the system for one time and this identity information can be used in different networks. Use of identity federation standards can not only help the user to across multiple networks include external networks with only one time log in, but also can help users from different networks to trust each other. Using identity federation in the cloud means users from different clouds can use a federated identification to identify themselves, which naturally suit the requirement of identity based cryptography in cloud computing. In our approach, users and servers in the cloud have their own unique identities. These identities are hierarchical identities. To access services in the cloud, users are required to authenticate themselves for each service in their own clouds. In some cases, servers are also required to authenticate themselves to users. In a small and closed cloud, this requirement can be satisfied easily. While in a hybrid cloud, there are multiple private and/or public clouds and these clouds may rely on different authentication mechanisms. Providing effective authentications for users and servers from different cloud domains would be difficult. In this paper, we propose to use federated identity management and HIBC in the cloud. In the cloud trusted authority PKGs are used and these PKGs will not only act as PKGs in traditional identity-based cryptography system but also allocate hierarchical identities to users in their domains. There is a root PKG in overall domain of each cloud, and each sub-level domain (private or public cloud) within the cloud also has its own PKG. The root PKG will manage the whole cloud, each private cloud or public cloud is the first level and users and servers in these clouds are the second level. The root PKG of the cloud will allocate and authenticate identities for all the private and public clouds. For example, it can allocate identity UiS to a private cloud of University of Stavanger. Each private cloud and public cloud uses its own domain PKG to allocate and manage the identities of all the users and servers in its own cloud. Each user and server in this domain has its own identity and this identity is a hierarchical identity, which includes both the identity of the user or server and the identity of the domain. For example, the identity of user Alice in the private cloud of University of Stavanger can be UIS. Alice.

4.2 Key Generation and in the Cloud

Using HIBC in the cloud, an important part is key generation and distribution. As shown in [6], the security of HIBC scheme is based on the using of admissible pairing. Let G1 and G2 be two groups of some large prime order q and G 1 is an additive group and G2 is a multiplicative group, we can call eˆ an admissible pairing if eˆ : GG G 12 2 × → have the following properties. 1. Bilinear: For all 1 PQ G, ∈ and, q ab Z ∗ ∈, ˆ ˆ (,) (,) ab e aP bQ e P Q =. 2. Non-degenerate: There exits 1 PQ G, ∈, such that ePQ ˆ (,) 1 ≠. 3. Computable: For all 1 PQ G, ∈, there exists an efficient way to calculate ePQ ˆ (,). An admissible pairing can be generated by suing a Weil pairing or a Tate pairing [2]. Here, in the cloud we use two levels PKG, the root PKG is 0 levels PKG and the PKGs in the private or public clouds are 1 level PKGs. The root setup can be done as follow: 1. Root PKG generates G 1 , G 2 and an admissible pairing 12 00 1 2 e aP bQ e P Q G G e P Q H H ˆˆ ˆ (,) (,) 1(, , , , , ,) = ≠ eˆ : GG G 12 2 × → . 2. Root PKG chooses P G 0 1 ∈ and 0 q s Z ∗ ∈ and set Q sP 0 00 =. 3. Root PKG chooses hash function H G 1 1: 0, 1 { } ∗ → and 2 2: 0, 1 { } n H G →. Then the system parameters are 12 00 1 2 (, ,, , , ,) G G eP Q H H ˆ and are public available, 0 s is the root PKG's secret and is known only by the root PKG. For the lower level PKGs and users and servers in the cloud, they can use the system parameters and any user's identity to generate its public key. And every user or servers in the cloud can connect the PKGs in their cloud domain to get their private keys. For example, the PKG in private cloud of University of Stavanger with identity UIS, its public key can be generated as 1 ( ) uis p H U IS = and the root PKG can generate its private key as uis uis 0 s sP =. For a user with identity UIS.Alice in the private cloud University of Stavanger , her public key can be generated as 1 ( ) P H UIS Alice uisalice = & and the PKG can generate her private key as uisalice uis uis uisalice s s sP = + .

4.3 Date Encryption and Digital Signature

In the cloud, one of the most important security problems are mutual authentication between users and servers, protection of data confidentiality and integrity during data transmission by encryption using secret keys. In a cloud using federated identity, any user and server has its unique identity and any user and server can get the identity of any other user/server by request with the PKGs. With HIBC, the public key distribution can be greatly simplified in the cloud. Users and servers do not need to ask a public key directory to get the public key of other users and servers as in traditional public key schemes. If any user or server wants to encrypt the data

that transmitted in the cloud, the sender can acquire the identity of the receiver, then the sender can encrypt the data with receiver's identity. Currently, WS-Security (Web service Security) protocol which can provide end-toend message level security using SOAP messages is widely applied in cloud computing to protect the security of most cloud computing related web services. WS-Security uses SOAP header element to carry security-related information. Since SOAP message is a kind of XML message and ordinarily XML message representation is about 4 to 10 times large compared with their equivalent binary formats, adding security information into SOAP header will greatly increase the costs of data communication and data parsing. For example, if XML signature is used to protect data integrity or authentication, the SOAP header will include the signature information about the signature method, signature value, key info and some reference information like digest method, transforms, and digest value. And the key info element may include keys names certificates and some public key management information [16]. If RSA and X.509 are chosen as the public key cryptography and certificate format in XML signature, the key info element in the SOAP header usually includes a public key certificate or a reference pointing to a remote location. While using HIBC in a cloud, any user and server can get its own private key from its domain PKG and can calculate the public key of any other party in the cloud knowing its identity. Then it is easy for a sender to add a digital signature using its private key and for a receiver to verify a digital signature using the sender's public key. Then the key info may be not needed in the SOAP header, and this will greatly reduce the SOAP messages need to be transmitted in the cloud and thus save the cost.

4.4 Secret Session Key Exchange and Mutual Authentication

Identity-based cryptography is a public key cryptography scheme; it is much slower when it is compared with symmetric key cryptography. In practice, public key cryptography is not used for data encryption in most of the clouds. For example, in XML encryption, XML data is encrypted using symmetric cryptography such as AES and Triple-DES. This secret symmetric key is encrypted using the public key encryption and added in the SOAP message and then transmitted to the receiver. While in the cloud with HIBC, this secret symmetric key distribution can be avoided since identity-based cryptography can be used for secret session key exchange. According to [9], for every two parties in the system using identity-based cryptography, it is easy for each one of the two parties to calculate a secret session key between them using its own private key and public key of other party, this is call identity-based noninteractive key distribution. For example, two parties Alice and Bob in a cloud with their public keys and private keys Palice, Qalice, Pbob and Qbob can calculate their shared secret session key by computing K eQ P eQ P s alice bob bob alice = = (1) This means in a cloud using HIBC, each user or server can calculate a secret session key between it and the other party it wants to communicate with without message exchange. This advantage of identity-based cryptography can not only reduce message transmission but also can avoid session key disclosure during transmission. This secret session key can be used not only for data encryption, but also for mutual authentication [8]. We assume if a user with identity Alice@UiS and a server with identity Storage@google in the cloud want to authenticate each other. First, they can calculate a secret session key K s between them. Then Alice can send a message to the server as: : @ , ,(, @ , @ ,) Alice Server Alice UiS M f K Alice UiS Storage google M → s Here M is a randomly selected message and f is a one way hash function. Here, to compute the correct hash value, a correct secret session key K s is needed. Since K s computation requires Alice's private key and this private key can only be allocated from the PKG in the private cloud of University of Stavanger, thus Alice can be verified that she is a legal user of this cloud. Also the server can authenticate itself to Alice the same way. We can notice that this mutual authentication does not include any certification form a third party.

5. CONCLUSION

The quick development of cloud computing bring some security problems as well as many benefits to Internet users. Current solutions have some disadvantages in key management and authentication especially in a hybrid cloud with several public/private clouds. In this paper, we depicted the principles of identity-based cryptography and hierarchical identity-based cryptography and find the properties of HIBC fit well with the security demands of cloud. We proposed to use federated identity management and HIBC in the cloud and depicted how can the system generate and distribute the public and private keys to users and servers. Compared with the current Ws-Security approach, we can see our approach has its advantages in simplifying public key distribution and reducing SOAP header size. Also we showed how the users and servers in the cloud can generate secret session key without message exchange and authenticate each other with a simple way using identity-based cryptography. Also we can see the key escrow problem of identity-based cryptography can be restricted with HIBC approach

6. REFERENCE

[1]Beak, J., Newmarch, J., Safavi-Naini, R., Susilo, W.: A Survey of Identity-Based Cryptography. In: Proc. of the 10th Annual Conference for Australian UNIX User's Group (AUUG 2004), pp. 95–102 (2004)

[2]Liang Yan "Federal Identity Management Using Hierarchical Identity-Based Cryptography", 2009

[3]Chappell, D.: A Short Introduction to Cloud Platforms, http://www.davidchappell.com/CloudPlatforms–Chappell.pdf

[4]Cocks, C.: An Identity-based Encryption Scheme Based on Quadratic Residues. In: Proceeding of 8th IMA International Conference on Cryptography and Coding (2001)

[5]Crampton, J., Lim, H.W., Paterson, and K.G.: What Can Identity-Based Cryptography Offer to Web Services? In: Proceedings of the 5th ACM Workshop on Secure Web Services (SWS 2007), Alexandria, Virginia, USA, and pp. 26–36. ACM Press, New York (2007)

[6]Gentry, C., Silverberg, A.: Hierarchical ID-Based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)

[7]Horwitz, J., Lynn, B.: Toward Hierarchical Identity-Based Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)

[8]Mao, W.: An Identity-based Non-interactive Authentication Framework for Computational Grids. HP Lab, Technical Report HPL-2004-96 (June 2004)

[9]Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems based on pairing. In: Proceedings of the 2000 Symposium on Cryptography and Information Security, Okinawa, Japan (January 2000)

[10]Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)

[11]Lim, H.W., Robshaw, M.J.B.: On identity-based cryptography and GRID computing. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, and J. (eds.) ICCS 2004. LNCS, vol. 3036, pp. 474–477. Springer, Heidelberg (2004)

[12]Lim, H.W., Paterson, K.G.: Identity-Based Cryptography for Grid Security. In: Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing (eScience 2005). IEEE Computer Society Press, Los Alamitos (2005)

[13]Defining Cloud Services and Cloud Computing, http://blogs.idc.com/ie/?p=190

[14]IBM Embraces Juniper for Its Smart Hybrid Cloud, Disses Cisco (IBM), http://www.businessinsider.com/2009/2/ibm-embraces-juniper-for-its-smart-hybrid-cloud-disses-cisco-ibm

[15]http://en.wikipedia.org/wiki/Cloud_computing#cite_note-61

[16] XML Signature Syntax and Processing (Second Edition), http://www.w3.org/TR/xmldsig-core/#sec-KeyInfo.

[17] Boneh, D., Franklin, M.: Identity-based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 433–439. Springer, Heidelberg (2001)