ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

PARALLEL NETWORK FILE SYSTEM USING pNFS -AKE KEY EXCHANGE FOR CLIENT/ SERVER MODEL

KARTHIGA.C,
PG Student,
Priyadarshini Engineering College, vaniyambadi-635751.
Karthigacse12@gmail.com.
PRAVEENA.G,
Associate Professor,
Priyadarshini Engineering College, Vaniyambadi-635751.
Cheziyan37@gmail.com.

ABSTRACT

Now a day's authentication of protocols is very important for secure many-many communication. The problem arises by proliferation of large scale die file system support the parallel access file system i.e. parallel Network File System (PNFS) the communication between clients to storage system may establish the session keys of Kerberos file system. The file system of Kerberos drawbacks are Meta data does not provide more security between client and server. Meta data servers generate themselves for all session key which lied between client and server. Meta data exchange exhibits heavy work load that restrict the scalability of protocols. In this paper we present a Varity of security key exchange protocols that are design to solve the above issue which arises. We show that our protocol have been reduced up to 60% of workload of the meta data server and concurrently support guarantee upon certain condition and forward secrecy and we improve performance with effectively.

I.INTRODUCTION

Authentication is the process of proving one's identity to someone else. As humans, we authenticate each other in many ways: We recognize each other's faces when we meet, we recognize each other's voices on the telephone.

An authentication protocol would run before the two communicating parties in the system run some other protocol. The authentication protocol first establishes the identity of the parties to each other's satisfaction; only after authentication do the parties get down to the work at hand. It is a fundamental building block for a secure networked environment.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

Kerberos [1] is an authentication service developed at MIT (Massachusetts Institute of Technology).that uses symmetric key encryption techniques and a key distribution centre; it is an add-system that can be used with existing network.

Kerberos provides a means of verifying the identities of principals on an open (unprotected) network. This is accomplished without relying on authentication by the host operating system, without basing trust on host address, without requiring physical security of all the hosts on the network, and under the assumption that packets travelling along the network can be read, modified, and inserted at will.

Kerberos performs authentication under these conditions as a trusted third party authentication service by using conventional cryptography. It trusted in the sense that each of its clients believes Kerberos's judgement as to the identity of each of its other clients to be accurate.

The problem that Kerberos addresses is this: a distributed system in which users at workstations wish to access services on servers distributed throughout the network. We would like for servers to be able to restricted access to authorized users and to be able to authenticate requests for service. In this system the following three threats exist:

1. A user may gain access to a particular workstation and pretend to be another user operating from that workstation.
2. A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.
3. A user may eavesdrop on exchanges and use a reply attack to gain entrance to a server or to disrupt operations.

In any of these cases, an unauthorized user may be able to gain access to services and data that he or she is not authorized to access. Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users.

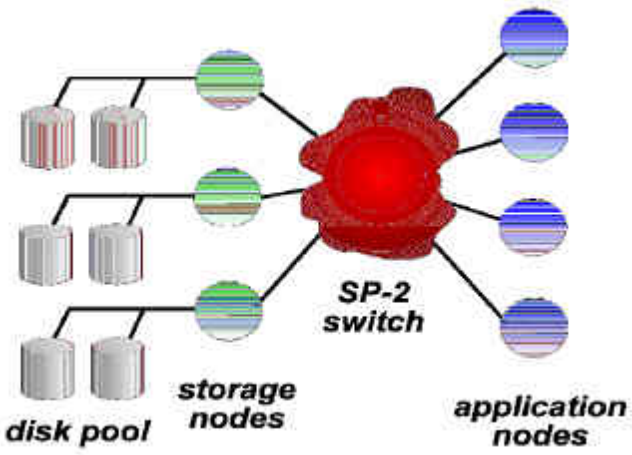
Kerberos provides the following requirements:

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

1. Secure: a network eavesdropper should not be able to obtain the necessary information to impersonate a user.
2. Reliable: Kerberos should be highly reliable and should employ a distributed server architecture, with one system able to back up another.
3. Transparent: the user should not be aware that the authentication is taking place, beyond the requirement to enter a password.
4. Scalable: the system should be capable of supporting large numbers of clients and servers.



There are three main types of nodes: file system, storage and manager. Any node can perform one of the above functions

File system node: co-ordinates administrative task

Manager nodes: exist one per file system. Different manager nodes are global lock manager, local lock manager, allocation manager etc

Storage node: implement shared access to files, co-ordinate with manager node during recovery and facilitate both file data and metadata to be striped across multiple storage nodes.

Some other auxiliary nodes include:

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

Metanode: A node is dynamically elected as metanode for centralized management of file metadata. Election of meta node is facilitated by *token server.*

Token Server: A token server tracks all tokens granted to all nodes in the cluster. A token granting algorithm is used to reduce cost of token management.

GPFS addresses the large file system issues are:

- High throughput
- High Availability
- Parallelism
- Fault-tolerance
- Scalability

Predominantly, we challenge to get together the subsequent advantageous possessions, which moreover have not been adequately accomplished or are not attainable by the present Kerberos based solution:

A. Scalability

The metadata server make possible right of entry requirements from a client to manifold storage devices be supposed to bear as modest workload as probable such that the server will not turn out to be a presentation bottleneck, but is competent of underneath incredibly great figure of clients;

B. Forward Secrecy

The procedure should assurance the safety of history session keys when the long-standing clandestine key of a client or a storage device is negotiation [10].

C. Escrow Free

The metadata server should not be taught any sequence about any assembly key used by the client and the storage device, make available there is no conspiracy in the middle of them.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

The main results of this paper are three new provably protected genuine key exchange procedures. Our procedures, more and more calculated to accomplish each of the above possessions, make obvious the trade-offs between competence and safety measures. We give you an idea about that our protocols can diminish the workload of the metadata server by in the region of half measured up to the current Kerberos based procedure, while achieving the desired security properties and observance the computational in the clouds at the clients and the storage devices at a rationally near to the ground level. We identify a proper refuge model and demonstrate that our procedures are protected in the representation.

II. RELKATED WORK

Telecare Medical Information Systems (TMIS) give a successful approach to enhance the restorative procedure between specialists, attendants and patients. By enhancing the security and protection of TMIS, it is vital while testing to enhance the TMIS so that a patient and a specialist can perform synchronized verification and session key foundation utilizing a 3-party therapeutic server while the safe information of the patient can be guaranteed. In proposed framework a mysterious three-party secret word validated key swapping (3PAKE) convention for TMIS is utilized. The convention depends on the proficient elliptic bend cryptosystem. For security, we apply the pi math based formal confirmation device ProVerif to demonstrate that our 3PAKE convention for TMIS can give namelessness to patient and specialist and also accomplishes synchronized verification and session key security.

The upside of proposed plan is security and effectiveness that can be utilized as a part of TMIS. For this J-PAKE based conventions are utilized. The disservice of proposed plan is of it lessened session keys. In proposed plan, two basic passwords based encoded key swapping conventions in light of that of Bellovin and Merritt. While one convention is more suitable to situations in which the secret key is shared over various servers, alternate gives better security. Both conventions are as productive, if worse, as any of the current scrambled key swapping conventions in the writing, but they just require a solitary arbitrary prophet case. The evidence of security for both conventions is in the irregular prophet show and in view of hardness of the

computational Diffe-Hellman issue. Passwords are a standout amongst the most well-known reasons for framework crashes, in light of the fact that the low entropy of passwords makes frameworks helpless against beast power speculating assaults. Because of new innovation passwords can be hacked effectively.

Robotized Turing Tests keep on being a viable, simple to-send way to deal with distinguish mechanized malevolent login endeavors with sensible expense of burden to clients. Subsequently in this proposed plan the insufficiency of existing and proposed login conventions intended to address substantial scale online lexicon assaults e.g. from a botnet of a huge number of hubs. In this plan proposed a basic plan that fortifies watchword based verification conventions and forestalls online word reference assaults and also numerous to-numerous assaults regular to 3-pass SPAKA conventions.

Proposed plan Uses compositional technique for demonstrating cryptographically solid security properties of key swapping conventions, in view of a typical rationale that is translated over ordinary keeps running of a convention against a probabilistic polynomial time aggressor. We show an automated confirmation of the secret word based convention One-Encryption Key Swapping (OEKE) utilizing the computationally-stable convention prover CryptoVerif. OEKE is a non-insignificant convention, and accordingly automating its verification gives extra certainty that it is right. This contextual investigation was likewise a chance to execute a few imperative expansions of CryptoVerif, helpful for demonstrating numerous different conventions.

We have to be sure stretched out CryptoVerif to bolster the computational Diffie-Hellman suspicion. Secret key Authenticated Key Swapping (PAKE) concentrates how to set up secure correspondence between two remote gatherings singularly taking into account their mutual watchword, without requiring a Public Key Infrastructure (PKI). Regardless of broad exploration in the previous decade, this issue stays unsolved. Patent has been one of the greatest brakes in sending PAKE arrangements practically speaking. Here, we apply it to take care of the PAKE issue, and demonstrate that the convention is zero-learning as it uncovers nothing aside from one-piece data: whether the supplied passwords at two sides are the same. With clear

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

favorable circumstances in security, our plan has practically identical efficiency to the EKE and SPEKE conventions

III. EXISTING SYSTEM

Previous descriptions of NFS listening carefully on straightforwardness and competence, and were intended to employment well on intranets and restricted networks. Afterward, the afterward versions aspire to get better access and presentation within the Internet environment. However, safekeeping has then become a greater concern. Among many other sanctuary issues, user and server authentication within an open, dispersed, and cross-domain environment are a difficult matter. Key management can be tedious and luxurious, but an important aspect in ensuring security of the system.

NFS (since version 4), therefore, has been mandating that implementations support end-to-end authentication, where a user (through a client) mutually authenticates to an NFS server. Moreover, consideration should be given to the integrity and privacy(confidentiality) of NFS requests and responses. The RPCSEC GSS framework is currently the core security component of NFS that provides basic security services. RPCSEC GSS allows RPC protocols to access the Generic Security Services Application Programming Interface (GSS-API). The latter is used to facilitate exchange of credentials between local and remote communicating parties, for example between a client and a server, in order to establish a security context.

The metadata server grants access permissions (to storage devices) to the client according to pre-defined access control lists (ACLs). The client's I/O request to a storage device must include the corresponding valid layout. Otherwise, the I/O request is rejected. In an environment where eavesdropping on the communication between the client and the storage device is of sufficient concern, RPCSEC GSS is used to provide privacy protection.

A. Parallel Sessions

Parallel secure sessions between the clients and the storage devices in the parallel Network File System (pNFS). The current Internet standard—in an efficient and scalable manner. This is similar to the situation that once the adversary compromises the long-term secret

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

key, it can learn all the subsequence sessions. If an honest client and an honest storage device complete matching sessions, they compute the same session key. Second, two our protocols provide forward secrecy: one is partially forward securing with respect to multiple sessions within a time period.

B. Authenticated Key Exchange

Our primary goal in this work is to design efficient and secure authenticated key exchange protocols that meet specific requirements of pNFS. The main results of this paper are three new provably secure authenticated key exchange protocols. We describe our design goals and give some intuition of a variety of pNFS authenticated key exchange6 (pNFS-AKE) protocols that we consider in this work.

C. Forward Secrecy

The protocol should guarantee the security of past session keys when the long-term secret key of a client or a storage device is compromised. However, the protocol does not provide any forward secrecy. To address key escrow while achieving forward secrecy simultaneously, we incorporate a Diffie- Hellman key agreement technique into Kerberos-like pNFS-AKE-I. However, note that we achieve only partial forward secrecy (with respect to v), by trading efficiency over security.

IV. PROPOSED SYSTEM

4.1 Kerberos PNFS:

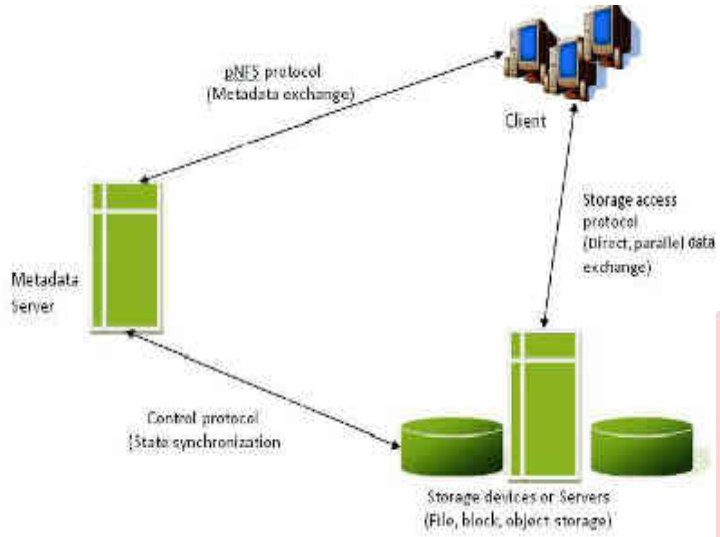
Step-1: $C \rightarrow M$: ID_C
Step-2: $M \rightarrow C$: $E(K_C;K_{CT})$, $E(K_T ; ID_C; t;K_{CT})$
Step-3: $C \rightarrow T$: $ID_{S_1}, \ldots, ID_{Sn}$, $E(K_T ; ID_C; t; K_{CT})$, $E(K_{CT} ; ID_C; t)$
Step-4: $T \rightarrow C$: $\boldsymbol{\sigma_1}, \ldots\ldots, \boldsymbol{\sigma_n}$, $E(\boldsymbol{K_{MS1}} ; \boldsymbol{ID_c}; t; \boldsymbol{SK_1})$; : : : ;$E(\boldsymbol{K_{MSn}}; \boldsymbol{ID_c}; t; \boldsymbol{SK_n})$, $E(\boldsymbol{K_{CT}} ; \boldsymbol{SK_1}, \ldots., \boldsymbol{SK_n})$
Step-5: $C \rightarrow S_i$: $\boldsymbol{\sigma_i}$;$E(\boldsymbol{K_{MSn}} ; \boldsymbol{ID_c}; t; \boldsymbol{SK_i})$, $E(\boldsymbol{SK_i}; \boldsymbol{ID_c}; t)$
Step-6: $S_i \rightarrow CS_i$: $E(\boldsymbol{SK_i}; t + 1)$

The Kerberos PNFS protocol can be split into three protocol techniques.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016



In this work, we research the issue of the protected numerous to numerous interchanges in the extensive scale system document frameworks which bolster parallel bring to various putting away gadgets. That we considering the correspondence model where there are an extensive number of the customers getting to numerous remote and conveyed stockpiling gadgets in parallel. Especially, we tries to concentrate on the most proficient method to swapping the key materials and foundation of the parallel secure sessions in the middle of customers and stockpiling gadgets in the parallel Network File System (pNFS), the present Internet norms in productive and adaptable way.

The advancement of pNFS is driven by Sun, EMC, IBM, and UMich/CITI, and subsequently it offers numerous comparative elements and is perfect with numerous current business system record frameworks. Our primary objective in this work is to outline effective and secure confirmed key swapping conventions that address particular issues of pNFS. Especially, we endeavor to meet the accompanying attractive properties, which have not been agreeably accomplished or are not achievable by current Kerberos-based arrangement.

4.1.1 PNFS AKE-I

Phase I – For each validity period *v*:

(1) $C \; ! \; M : ID_C, E(K_{CM}, K_{CS_1}; : : : ; K_{CS_N})$
(2) $M \; ! \; C : E(K_{MS_1}; ID_C; ID_{S_1}; v; K_{CS_1}); : : : ; E(K_{MS_N}; ID_C; ID_{S_N}; v; K_{CS_N})$

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

Phase II – For each access request at time *t*:

(1) $C \rightarrow M : ID_C, ID_{S1}, : : : , ID_{Sn}$
(2) $M \rightarrow C : \_1, : : : , \_n$
(3) $C \rightarrow S_i : \_i, E(K_{MSi}, ID_C, ID_{Si}, v, K_{CSi}), E(sk_{0i}, ID_C, t)$
(4) $S_i \rightarrow C : E(sk_{0i}, t + 1)$

4.1.2 PNFS AKE-II

Phase I – For each validity period *v*:

(1) $S_i \rightarrow M : ID_{Si}, E(K_{MSi}, g_{si})$
(2) $C \rightarrow M : ID_C, E(K_{CM}, g_c)$
(3) $M \rightarrow C : E(K_{CM}, g_{s1}, : : : , g_{sN}), \_ (K_{MS1}, ID_C, ID_{S1}, v, g_c, g_{s1}), : : : , \_ (K_{MSN}, ID_C, ID_{SN}, v, g_c, g_{sN})$

Phase II – For each access request at time *t*:

(1) $C \rightarrow M : ID_C, ID_{S1}, : : : , ID_{Sn}$
(2) $M \rightarrow C : \_1, : : : , \_n$
(3) $C \rightarrow S_i : \_i, g_c, \_ (K_{MSi}, ID_C, ID_{Si}, v, g_c, g_{si}), E(sk_{0i}, ID_C, t)$
(4) $S_i \rightarrow C : E(sk_{0i}, t + 1)$

4.1.3 PNFS AKE-III

Phase I – For each validity period *v*:

(1) $S_i \rightarrow M : ID_{Si}, E(K_{MSi}, g_{si})$
(2) $C \rightarrow M : ID_C, E(K_{CM}, g_c)$
(3) $M \rightarrow C : E(K_{CM}, g_{s1}, : : : , g_{sN})$
(4) $M \rightarrow S_i : E(K_{MSi}, ID_C, ID_{Si}, v, g_c, g_{si})$

Phase II – For each access request at time *t*:

(1) $C \rightarrow M : ID_C, ID_{S1}, : : : , ID_{Sn}$
(2) $M \rightarrow C : \_1, : : : , \_n$
(3) $C \rightarrow S_i : \_i, E(sk_{j,0i}, ID_C, t)$
(4) $S_i \rightarrow C : E(sk_{j,0i}, t + 1)$

4.2 Security Analysis

4.2.1security model

We work in a security model that allows us to show that an adversary attacking our protocols will not able to learn any information about a session key. Our model also implies

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

implicit authentication, that is, only the right protocol participant is able to learn or derive a session key.

1. SEND(P; i;m)
2. CORRUPT(P)
3. REVEAL(P; i)
4. TEST($P*$; $i*$)

4.2.2security proofs

The security proofs mainly contain 3 theorems they are

Theorem 1: The pNFS-AKE-I protocol is secure without PFS if the authenticated encryption scheme E is secure under chosen-ciphertext attacks and F is a family of pesudo-random functions.

Theorem 2: The pNFS-AKE-II protocol achieves partial forward secrecy if T is a secure MAC scheme, the DDH assumption holds in the underlying group G, and F is a family of pesudo-random functions.

Theorem 3: The pNFS-AKE-III protocol achieves full forward secrecy if E is a secure authenticated encryption scheme, the DDH assumption holds in the underlying group G, and F is a family of pesudo-random functions.

4.2.3 performance Evaluation

1.Kerberos-pNFS

a) Symmetric key encryption / decryption= $w(6n + 8)$

b) MAC generation / verification =$2wn$

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

***2.* pNFS-AKE-I**

a) Symmetric key encryption / decryption = $5wn + N + 2$
b) MAC generation / verification = $2wn$
c) Key derivation = $4wn$

***3.* pNFS-AKE-II**

a) Symmetric key encryption / decryption = $4wn + N + 5$
b) MAC generation / verification = $3wn + N$
c) Key derivation = $4wn$
d) Diffie-Hellman exponentiation = $2N + wn + 1$

4. pNFS-AKE-III

a) Symmetric key encryption / decryption = $4wn + 2N + 5$
b) MAC generation / verification = $2wn$
c) Key derivation = $6wn + 2N$
d) Diffie-Hellman exponentiation = $3N + 1$

V.CONCLUSION

Our protocols offer three appealing advantages over the existing Kerberos-based pNFS protocol. First, the metadata server executing our protocols has much lower workload than that of the Kerberos-based approach. Second, two our protocols provide forward secrecy: one is partially forward secure (with respect to multiple sessions within a time period), while the other is fully forward secure (with respect to a session). Third, we have designed a protocol which not only provides forward secrecy, but is also escrow-free.

VI.REFERANCE

[1] M. Abd-El-Malek, W.V. Courtright II, C. Cranor, G.R. Ganger, J. Hendricks, A.J. Klosterman, M.P. Mesnier, M. Prasad, B. Salmon, R.R. Sambasivan, S. Sinnamohideen, J.D. Strunk, E. Thereska, M. Wachs, and J.J. Wylie. Ursa Minor: Versatile cluster-based storage. In *Proceedings of the 4th USENIX Conference on File and Storage Technologies (FAST)*, pages 59–72. USENIX Association, Dec 2005.

[2] A. Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J.R. Douceur, J. Howell, J.R. Lorch, M. Theimer, and R. Wattenhofer. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. In *Proceedings of the 5th Symposium on Operating System Design and Implementation (OSDI)*. USENIX Association, Dec 2002.

[3] M.K. Aguilera, M. Ji, M. Lillibridge, J. MacCormick, E. Oertli, D.G. Andersen, M. Burrows, T. Mann, and C.A. Thekkath. Blocklevel security for network-attached disks. In *Proceedings of the 2nd International Conference on File and Storage Technologies (FAST)*. USENIX Association, Mar 2003.

[4] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Communications of the ACM*, 53(4):50–58. ACM Press, Apr 2010.

[5] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Advances in Cryptology – Proceedings of CRYPTO*, pages 258–275. Springer LNCS 3621, Aug 2005.

[6] R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *Advances in Cryptology – Proceedings of EUROCRYPT*, pages 453–474. Springer LNCS 2045, May 2011.

[7] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In *Proceedings of the 6th Symposium on Operating System Design and Implementation (OSDI)*, pages 137–150. USENIX Association, Dec 2004.

[8] S. Emery. Kerberos version 5 Generic Security Service Application Program Interface (GSS-API) channel binding hash agility. *The Internet Engineering Task Force (IETF)*, RFC 6542, Mar 2012.