Certificate based key exchange scheme for encrypted communication in NTMobile networks

Yuya Miyazaki†, Fumihito Sugihara*, Katsuhiro Naito†, Hidekazu Suzuki‡, Akira Watanabe‡,
†Faculty of Information Science, Aichi Institute of Technology, Toyota, Aichi 470-0392, Japan
Email: miyazaki121@pluslab.org, naito@pluslab.org
*Department of Electrical and Electronic Engineering, Mie University, Tsu, Mie 514-8507, Japan
‡Graduate School of Science and Technology, Meijo University, Nagoya, Aichi 468-8502, Japan

***Abstract*—Smartphones and Internet of Things (IoT) devices introduce a new type of application such as end-to-end realtime services. End-to-end communication is suitable for these services because both end-nodes should exchange information with each other. On the contrary, almost all systems employ a client server system because practical networks have some limitations due to a NAT traversal, a different protocol version, multi-home networks etc. IP mobility and accessibility network protocols are a useful technology to realize end-to-end communication. The authors have proposed a new protocol that is called Network Traversal with Mobility (NTMobile). NTMobile realizes end-to-end communication in IPv4 and IPv6 networks and solves NAT traversal. It also serves a secure key distribution mechanism between NTMobile nodes for secure communication. However, requirements for security are increasingly-demanded during recent years. This paper proposes a new secure mechanism based on certificates for NTMobile. The proposed mechanism provides certificates to each NTMobile nodes. NTMobile nodes can verify a certificate to confirm the validity of each NTMobile nodes. Additionally, they can exchange an encryption key for secure communication between NTMobile nodes directly. In the prototype verification, we employ openSSL to apply the proposed mechanism. The verification demonstrates that the overhead of the proposed mechanism is quite insignificant in practical use cases.**

I. Introduction

Smartphones and Internet of Things (IoT) devices introduce a new type of application such as end-to-end realtime services. Some smartphone applications realize a chat service, a voice communication service, a realtime game, etc. IoT devices can perform collaborative operations between IoT devices by exchanging information with each other. End-to-end communication is suitable for these services because both end-nodes should exchange information with each other. On the contrary, almost all systems employ a client server system because practical networks have some limitations due to a NAT traversal, a different protocol version, multi-home networks etc. Additionally, real time communication services typically prefer continuous communication. However, the communication may be disconnected when operating systems change wireless interfaces due to client movement or access policies.

IP mobility and accessibility network protocols are a useful technology to realize end-to-end communication [1], [2], [3], [4]. Some implementations for IPv6 have been proposed as IP mobility mechanisms. On the contrary, practical implementations for IPv4 are quite rare though some mechanisms have been proposed [5], [6], [7]. IPv4 is generally used in Internet provider networks because IPv4 is the mainstream protocol in the current Internet. Additionally, a large scale network address translator (LSN) may be employed to meet the shortage of IPv4 global addresses [8], [9], [10]. As a result, IP mobility in the case when NAT exists between end nodes is also an important issue.

The authors have proposed a new protocol that is called Network Traversal with Mobility (NTMobile)[11], [12]. NTMobile realizes end-to-end communication in IPv4 and IPv6 networks and solves NAT traversal. It also serves a secure key distribution mechanism for NTMobile nodes. Therefore, secure communication between NTMobile nodes is still supported. However, requirements for security are increasingly-demanded during recent years. Hence, NTMobile should also support more strict secure communication to realize certification of NTMobile nodes and a distributed key exchange mechanism.

This paper proposes a new secure mechanism based on certificates for NTMobile. We extend NTMobile protocol to support the proposed secure mechanism. The proposed mechanism provides certificates to each NTMobile nodes. NTMobile nodes can verify a certificate to confirm the validity of each NTMobile nodes. Additionally, they can exchange an encryption key for secure communication directly. In the prototype verification, we employ openSSL[13] to apply the proposed mechanism. The verification demonstrates that the proposed mechanisms can work with the signaling process in NTMobile, and the overhead of the proposed mechanism is not an issue in practical use cases.

II. NTMobile

Fig. 1 shows the overview of NTMobile. NTMobile is IP mobility and accessibility network protocol for IPv4 and IPv6 networks, and can realize an overlay network where every NTMobile nodes can connect with each other flexibly. NTMobile system consists of Account Server (AS), Direction Coordinators (DC), Relay Server (RS), Cache Server (CS), and NTMobile nodes. Each NTMobile node can be recognized by its FQDN which is assigned by its DC. Additionally, NTMobile nodes also realize continuous communication with virtual IP addresses corresponding to their FQDN. Details of NTMobile components are as following.

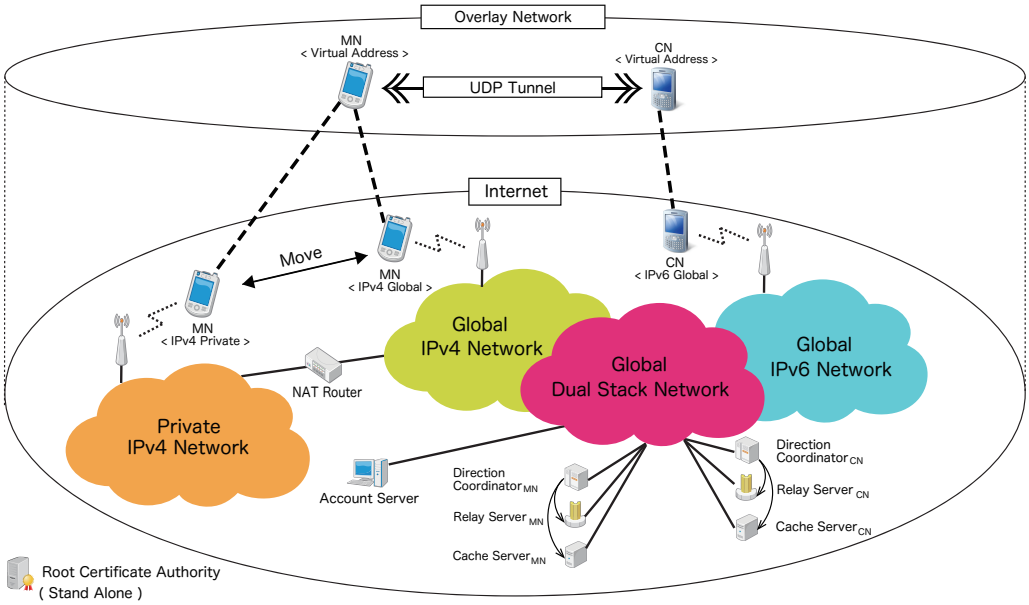

Fig. 1. System model of NTMobile.

- Account Server
  AS is an only equipment that manages authentication information for NTMobile nodes. AS replies an authentication reply when a DC makes inquiries about their end-nodes' authentication to AS.
- Direction Coordinator
  The function of Direction Coordinator is to manage location information of each NTMobile node. They also assign a virtual IP address and FQDN to each NTMobile nodes. In addition, they indicate signaling processes for tunnel creation between MN and CN. Each DC has the DNS function and, its FQDN is registered in Name Server (NS) record. Therefore, DC can easily find another DC by querying NS records.
- Relay Server
  Relay servers relay a tunnel between NTMobile nodes when they cannot communicate directly due to a NAT traversal and difference of the IP protocol version.
- Cache Server
  The function of cache servers is to store exchanged data for NTMobile nodes in the non-realtime secure data exchange mechanisms. CN can download the stored data from CS at anytime because CS can store the uploaded data from MN.

III. Key Exchange Scheme with Certificate

This paper proposes a certificate based key exchange scheme, where a NTMobile node can generate a shared key for communication and can exchange it between NTMobile nodes safely. The proposed scheme employs the public key infrastructure (PKI) [14], [15], [16]. Hence, a certificate chain of the proposed scheme assumes an original Root

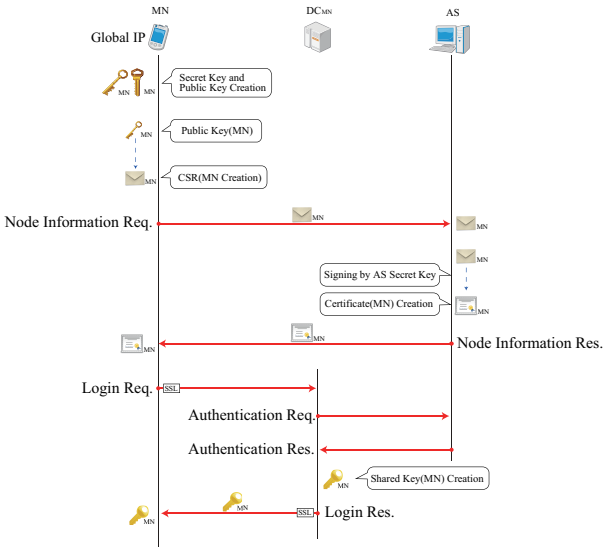

Fig. 2. Login process.

Certificate Authority (RCA) which is called Root-CA. Root-CA can issue multiple certificates in the NTMobile domain. Therefore, Root-CA should sign each certificate for servers: account servers, direction coordinators, relay servers, and cache servers. The proposed key exchange scheme requires a certificate for each NTMobile node. It also assumes that an account server signs a certificate of each NTMobile node when the NTMobile node requests an authentication. Therefore, a mobile node (MN) can verify a certificate of a correspondent node (CN).

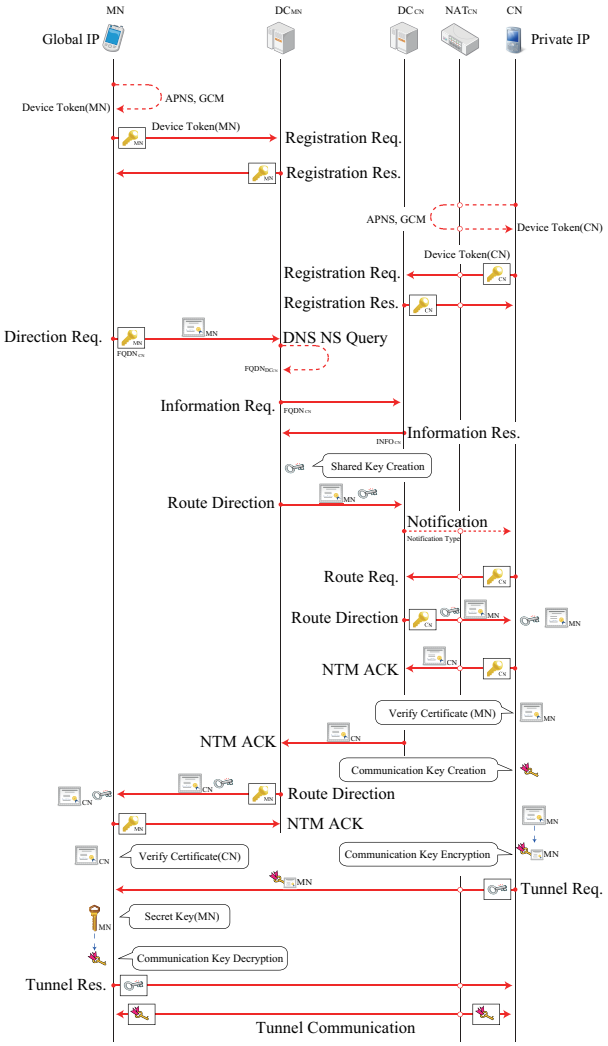

Fig. 3. Key exchange process (Direct).

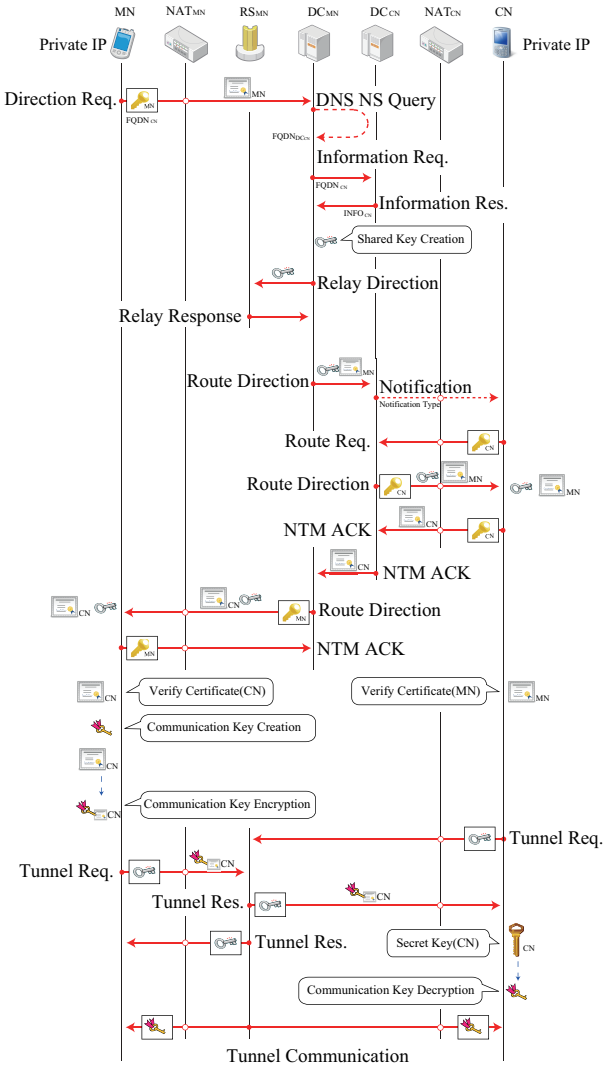

Fig. 4. Key exchange process (Relay).

A. *Login process*

Fig. 2 shows the login process. MN generates a public key and a secret key by a public key cryptosystem. In addition, MN creates a Certificate Signing Request (CSR) according to the public key. Then, it sends the node information request message to AS. The node information request message contains the CSR and the authentication information of MN. AS signs the CSR to create the certificate for MN when the received node information request message is authenticated. It also replies the node information response message to MN. The node information response message includes the certificates for MN. Then, MN transmits the login request message to $\mathrm{DC_{MN}}$. $\mathrm{DC_{MN}}$ transmits the login request message to AS when the received login request message is authenticated. $\mathrm{DC_{MN}}$ generates a shared key for communication with MN when received authentication message. Then, $\mathrm{DC_{MN}}$ transmits the login response message with the shared key to MN. The shared key is used for encryption between MN and $\mathrm{DC_{MN}}$.

B. *Key exchange process*

Fig. 3 shows the key exchange process in direct communication and Fig. 4 shows the key exchange process with relay operation by $\mathrm{RS_{MN}}$. The following is the signaling process of the key exchange mechanisms based certificates.

- $\mathrm{DC_{MN}}$ manages the location information of MN and $\mathrm{DC_{CN}}$ manages that of CN in NTMobile. The location information is updated by the registration request messages.
- MN transmits the direction request message to $\mathrm{DC_{MN}}$ to request the tunnel creation between MN and CN because MN should obtain the information about CN through both DCs.
- Each DC has the DNS function to manage its domain. Therefore, $\mathrm{DC_{MN}}$ can find $\mathrm{DC_{CN}}$ by transmitting the

DNS NS query.

- $\mathrm{DC_{MN}}$ transmits the Information Request message to $\mathrm{DC_{CN}}$ to obtain the information of CN.
- $\mathrm{DC_{CN}}$ transmits the Information Response message to $\mathrm{DC_{MN}}$ to notify the latest information of CN.
- $\mathrm{DC_{MN}}$ decides the communication route: a direct route or a relay route, according to the information of both MN and CN. In Fig. 3, the tunnel is created directly between MN and CN. In Fig. 4, the tunnel is relayed at $\mathrm{RS_{MN}}$.
- In Fig. 4, $\mathrm{DC_{MN}}$ transmits the relay direction message to the relay sever to request the relay operation of the tunnel communication between MN and CN. The relay direction message contains the shared key for encryption of message headers of NTMobile messages . $\mathrm{RS_{MN}}$ replies the relay direction response message to $\mathrm{DC_{MN}}$ when the relay operation is ready.
- $\mathrm{DC_{MN}}$ transmits the route direction message to $\mathrm{DC_{CN}}$ to notify the tunnel creation request to CN. The route direction message contains the shared key for encryption of message headers of NTMobile messages and the certificate of MN. $\mathrm{DC_{CN}}$ also notifies CN by notification mechanisms.
- CN transmits the route direction request message to $\mathrm{DC_{CN}}$ when it receives the notification from $\mathrm{DC_{CN}}$
- $\mathrm{DC_{CN}}$ transmits the route direction message to CN. The route direction message also contains the shared key for encryption of message headers and the certificate of MN.
- CN transmits the ACK message to $\mathrm{DC_{CN}}$. The ACK message contains the certificate of CN.
- $\mathrm{DC_{CN}}$ transmits the ACK message including the certificate of CN to $\mathrm{DC_{MN}}$.
- $\mathrm{DC_{MN}}$ transmits the route direction message to MN. The route direction message involves the shared key for encryption of message headers and the certificate of CN.
- MN transmits the ACK message to $\mathrm{DC_{MN}}$.
- The tunnel creation process requires nodes in global networks to receive a tunnel request message because nodes in private networks is not available from the global Internet. In Fig. 3, CN should transmits the tunnel request message to MN. The tunnel request message involves the communication shared key encrypted by the certificate of MN.
- MN transmits the tunnel response message to CN when it receives the tunnel request message and decrypts the communication shared key with own private key.
- In Fig. 4, both MN and CN should transmit the tunnel request messages to $\mathrm{RS_{MN}}$ to create the tunnels. The tunnel request messages include the communication shared key encrypted by the certificate of CN. Therefore, $\mathrm{RS_{MN}}$ can check only the message header and cannot decode the communication shared key. $\mathrm{RS_{MN}}$ replies the tunnel response messages to MN and CN respectively. The tunnel response messages involve the communication shared key encrypted by the certificate of CN.
- Both MN and CN share the communication shared key, and start the tunnel communication.

IV. Evaluation with openSSL

A. Evaluation setup

OpenSSL is a well known open source toolkit that implements the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS) protocols as well as a full-strength general purpose cryptography library. The proposed scheme also employs OpenSSL library to realize a secure key exchange mechanism. Therefore, we perform a experimental evaluation of the proposed scheme with OpenSSL library. We employ Rivest Shamir Adleman (RSA) and Advanced Encryption Standard (AES) as the cryptographic algorithm. Detail evaluation processes are described as follows.

- Issue of a certificate for Root-CA
  Root-CA generates a private key and a public key, and creates its CSR. Then, it also signs its own CSR by its private key. As a result, Root-CA becomes an original Root Certificate Authority.
- Issue of a certificate for AS
  AS generates a private key and a public key, and creates its CSR. Root-CA signs the AS's CSR by its private key. Therefore, AS becomes an intermediate certificate authority.
- Issue of a certificate for NTMobilde node
  NTMobile nodes generate a private key and a public key, and create their CSR. AS signs the CSR of NTMobile nodes by its private key.
- Verification of a certificate
  MN and CN verify a certificate each other with the certificates of Root-CA and AS.
- Exchange of an encryption key with certificates
  MN and CN create a communication shared key for AES, and encrypt it with a public key in the certificate. They decrypt the encrypted communication shared key with their own secret key.

B. Performance evaluation

Fig. 5 and Fig. 6 show the evaluation environment. We have set up three virtual servers for AS, MN and CN. We employed a 2048 [bit] key for RSA and a 256 [bit] key for AES. Detail evaluation procedure is described as follows.

- Generation of RSA key
  MN and CN generate their 2048 [bit] RSA key. Then, they create a CSR from the RSA key.
- Sign of certificate
  AS signs the CSR by the AS's secret key.
- Verification of certificate
  MN and CN obtain the Root-CA's certificate and the AS's certificate. Then, they verify another node's certificate by the CA's certificate.
- Generation of AES key
  MN and CN generate a 256 [bit] AES key.
- Encryption with AES key
  MN and CN encrypt the AES key by another node certificate.

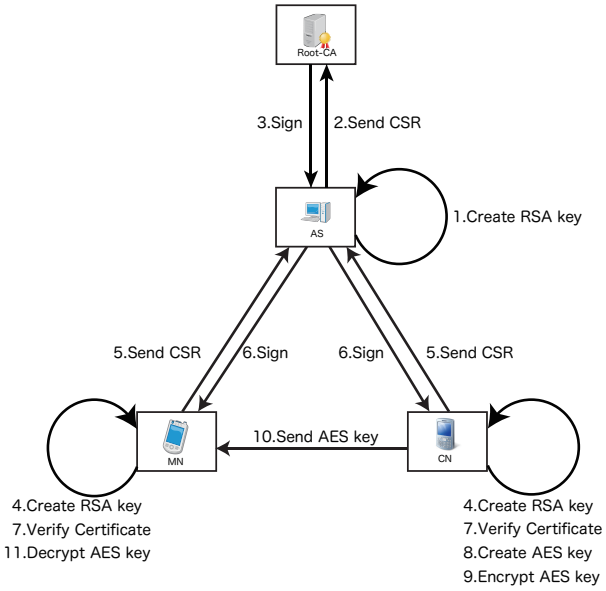

Fig. 5. Evaluation process.

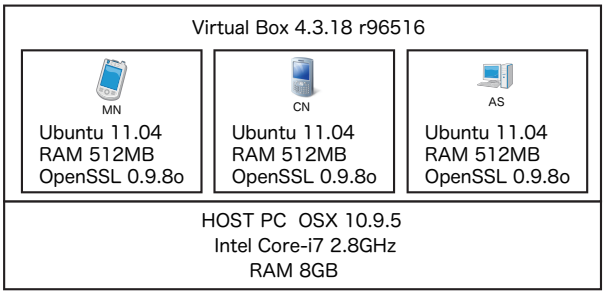

Fig. 6. Specification of device.

- Decryption with AES key
  MN and CN decrypt the AES key by their own secret key.

Tab. I shows the evaluation results about the processing period of each process. From the results, the key generation process of RSA takes more than 1 [s]. The RSA key is generated at only a first login process. Therefore, the key generation period for RSA is not big issues in practical situations. Additionally, the overhead of the tunnel creation is lower than 0.1 [s]. As a result, the proposed scheme can perform secure communication without substantial overhead.

V. Conclusion

This paper has proposed a new secure mechanism based on certificates for NTMobile. The proposed mechanism provides certificates to each NTMobile nodes. NTMobile nodes can verify a certificate to confirm the validity of each NTMobile nodes. Additionally, they can exchange a communication encryption key between them directly. The verification shows that the proposed mechanisms can work with openSSL library.

TABLE I
Processing time.

Machine	Process	Time [s]
MN	Creating CSR	1.403
AS	Signing CSR	0.094
MN	Verifying certificate	0.007
MN	Generating AES key	0.007
MN	Encrypting with AES key	0.014
CN	Decrypting with AES key	0.051

Acknowledgment

This work is supported in part by Grant-in-Aid for Scientific Research (B)(15H02697) and Grant-in-Aid for Scientific Research (C)(26330103), Japan Society for the Promotion of Science (JSPS) and the Integration research for agriculture and interdisciplinary fields, Ministry of Agriculture, Forestry and Fisheries, Japan.

References

[1] D. Le, X. Fu and D. Hogrere, “A Review of Mobility Support Paradigms for the Internet,” IEEE Communications surveys, 1st quarter 2006, Volume 8, No. 1, 2006.
[2] C. Perkins, “IP Mobility Support for IPv4, Revised,” RFC 5944, IETF (2010).
[3] M. Ishiyama, M. Kunishi, K. Uehara, H. Esaki, and F. Teraoka, “LINA: A New Approach to Mobility Support in Wide Area Networks,” IEICE Trans. Comm., Vol.E84-B, No.8, pp.2076–2086, 2001.
[4] http://www.mip4.org, retrieved: February , 2012.
[5] S. Salsano, C. Mingardi, S. Niccolini, A. Polidoro and L. Veltri ”SIP-based Mobility Management in Next Generation Networks,” IEEE Wireless Communication, Vol. 15, Issue 2, April 2008.
[6] M. Bonola, S. Salsano and A. Polidoro, “UPMT: universal per-application mobility management using tunnels,” In Proc. of the 28th IEEE conference on Global telecommunications (GLOBECOM’09), December 2009.
[7] M. Bonola and S. Salsano, “S-UPMT: a secure Vertical Handover solution based on IP in UDP tunneling and IPsec,” GTTI Riunione Annuale 2010, (online), http://www.gtti.it/GTTI10/papers/gtti10_submission_29.pdf, 2010.
[8] E. Fogelstroem, A. Jonsson, and C. Perkins, “Mobile IPv4 Regional Registration,” RFC 4857, June 2007.
[9] H. Levkowetz and S. Vaarala, “Mobile IP Traversal of Network Address Translation (NAT) Devices,” RFC 3519, April 2003.
[10] G. Montenegro, “Reverse Tunneling for Mobile IP, revised,” RFC 3024, January 2001.
[11] K. Naito, K. Kamienoo, T. Nishio, T. Mizutani, H. Suzuki, A. Watanabe, K. Mori and H. Kobayashi, “Realization and Implementation of IP Mobility in NTMobile,” IPSJ Journal, Vol. 54, No. 1, pp. 380–393, January 2013.
[12] K. Naito, K. Kamienoo, H. Suzuki, A. Watanabe, K. Mori, and H. Kobayashi, “End-to-end IP mobility platform in application layer for iOS and Android OS,” IEEE Consumer Communications and Networking Conference (CCNC 2014), January 2014.
[13] https://www.openssl.org, retrieved: May, 2015.
[14] S. Berkovits, S. Chokhani, J. Furlong, and Jisoo, “Public Key Infrastructure Study: Final Report,” Produced by the MITRE Corporation for NIST, April 1994.
[15] J. Foley, “Provisioning X.509 certificates using RFC 7030,” Linux Journal archive, Vol. 2014 No. 245, September 2014.
[16] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” RFC 5280, May 2008.