

Using Multiple Cryptographic Modules for Server-based Certificate Validation System

Takahiro Fujishiro***, Akane Sato*, Yoko Kumagai*, Shingo Hane*, Tadashi Kaji*, Kenichi Okada**

Systems Development Laboratory, Hitachi, Ltd., Yokohama Japan

***Keio University, Yokohama Japan*

takahiro.fujishiro.zj@hitachi.com, akane.sato.mk@hitachi.com, yoko.kumagai.su@hitachi.com, shingo.hane.vz@hitachi.com, tadashi.kaji.ck@hitachi.com, okada@z2.keio.jp

Abstract— This paper propose an algorithm using multiple cryptographic modules to speed up a certification path validation of the X.509 Certificate. The algorithm used not only software cryptographic module but also hardware cryptographic modules in order to speed up transactions. The effect of it was confirmed by the experiment of the server system that had the fast algorithm of certification path validation under the private test environment.

Keywords—PKI, X.509, Certificate, Certification Path, Path Validation

I. INTRODUCTION

PKI (Public Key Infrastructure) is an infrastructure for data encryption, authentication and digital signature and it is defined in ITU-T Recommendation X.509[1]. It is used as SSL or TLS protocols [2,3]. Due to the development of legal systems so-called Electric Signature Law in many countries, PKI is also used as electric signature for concluding contracts and for authentication of users[4,5]. And PKI is applied in many other cases such as using for IEEE 802.1X network authentication and device authentication etc. Because of the increase of PKI application and utilized opportunities, a lighter loading and higher speed PKI operation method is demanded by the application users.

In PKI, a trusted third party, called CA (Certification Authority), certifies the relationship between certain public key and its owner and issues public key certificate. And CA also provides a revocation information of the certificate with issuing a CRL (Certificate Revocation List), ARL (Authority Revocation List), or serving a real-time revocation information using a OCSP (Online Certificate Status Protocol) [11] server.

Cryptographic processing as generating and verifying a signature value is a necessary operation to use PKI. In general, cryptographic operations are heavy process and need much computing resources. We hereunder describe about development of fast algorithm to validate a certificate by using multiple cryptographic modules.

II. VERIFYING CERTIFICATE

When authenticating by using PKI, it is necessary to verify the signature value and the certificate. To verify a value of a PKI digital signature, the public key recorded in the certificate is used for verification. To verify a certificate, it is necessary to confirm whether the certificate is issued correctly by a trusted CA. And a certificate would be revoked in case of a change in the certificate record or the private key compromise. Therefore a status of a certificate validity needs to be confirmed even it was issued correctly. If only verification of a digital signature value is executed and validation of the corresponding certificate isn't executed, it is impossible to detect an incorrect or revoked certificate and thus assurance of authenticity by PKI cannot be realized. Considering the above, it is important to verify certificates as well as to verify digital signatures when authenticating by using PKI. There are two validation approaches of certificates; the multiple trust anchors model and the cross certificate model. First, we describe about the multiple trust anchors model and the cross-certification model, and next, describe the method of certificate validation under the cross certification model and the method of processing it under the client server system.

A. Multiple Trust Anchors Model

In the multiple trust anchors model, the relying party holds a

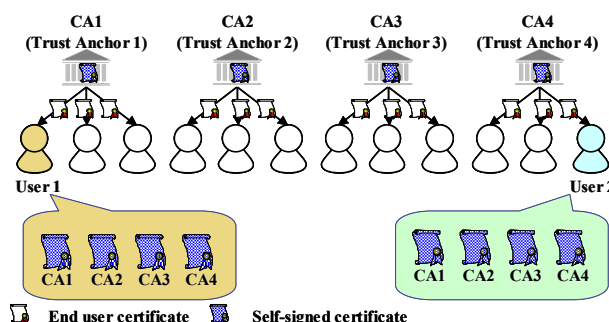


Figure 1. Multiple Trust Anchors model.

set of multiple CA certificates (i.e., self signed certificate). In other words, the relying party has multiple trust anchors (TAs). Current WWW (World Wide Web) adopts this model for server authentication of TLS (Transport Layer Security) protocol.

Figure 1 shows an example of multiple trust anchors model. In this figure, CA1, CA2, CA3 and CA4 are trust anchors, respectively. And user 1 and user 2 have four CA certificates.

When processing a certificate validation, several CA certificates are stored in the validation system as TAs. Multi CAs are trusted by users and their certificates are distributed and stored on validation system to be used for validation of end users certificate. It is the model used by ordinary web browsers to validate SSL server certificate.

B. Cross-Certification model

On the other hand, in the cross-certification model, the relying party has only one CA certificate as its trust anchor. And other CAs can be relied if those CAs has cross-certification with the trust anchor. The cross-certification between certain CA and the trust anchor can be established if the trust anchor issues a cross certificate to the CA or if another CA, which has the cross-certification with the trust anchor, issues a cross certificate to the CA.

Figure 2 shows an example of cross-certification model. In this figure, CA1 issues a cross certificate to CA2, CA2 issues cross certificates to CA1 and CA3, CA3 issues cross certificates to CA2 and CA4, and CA4 issues a cross certificate to CA3. And user 1 has CA1 certificate and user 2 has CA4 certificate.

In the ubiquitous network, various entities communicate with each other for the various purposes. And many terminals in the ubiquitous network will have less computing power and memory than current PC. In addition, some terminals in the ubiquitous environment will have difficulty of maintenance because these terminals will be widely distributed. On such ubiquitous network, the increase of terminals brings the increase of CAs. Therefore, the terminal has to maintain huge number of CA certificates in multiple trust anchors model. It means that the multiple trust anchors model is difficult to use in the ubiquitous network.

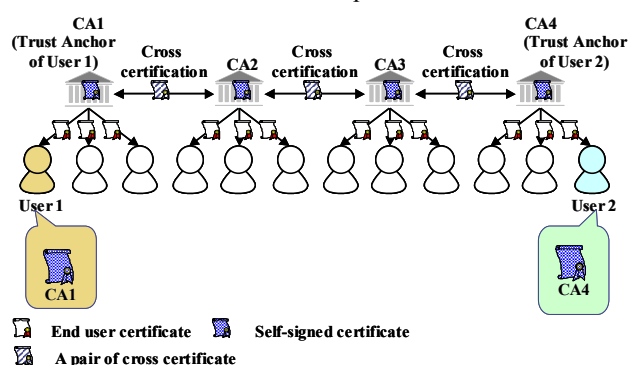


Figure 2. Cross-certification model.

In general, the cross-certification model requires smaller memory to hold trust anchor certificates than the multiple trust anchors model, though the cross-certification model takes more computing power than the multiple trust anchors model. For example, if there are millions of CAs in the ubiquitous network, the multiple trust anchors model requires to hold millions of CA certificates. But it is not needed to discover a certification path. On the other hand, the cross-certification model requires traversing a graph, in which nodes are CA certificates and edges are cross-certificates, to discover a certification path.

It is the model adopted by public sector such as Governmental Public Key Infrastructure (GPKI, LGPKI) [6,7] and the Public Certification Service for Residents in Japan (JPKI) [8] as shown in figure 3.

C. Certificate Validation Method

A certificate chain made of certificates is called a certification path. And a certification path is verified as a certificate validation under the cross certification model. Figure 4 shows an example of building a certification path to validate an End Entity (EE) certificate as a subscriber certificate. The certificate validation function builds a certification path from TA certificate to EE certificate and validates it. In this example, there are three cross certificates and four Certificate Revocation List (CRL)/Authority Revocation Lists (ARLs) to be used as certification information to validate the EE certificate. Each certificate and CRL/ARLs are searched and collected from the directory server called repository.

D. Delegated Path Validation Service

As a certificate validation under the cross certification model needs many processes, it is heavy loading for many client systems to validate all certificates in a certification path every time they receive it [10]. So it is considered effective to operate a certification validation processing at a server to reduce loading of client, and thus the protocols between a client and a server and the validation method are investigated [9,12]. As the

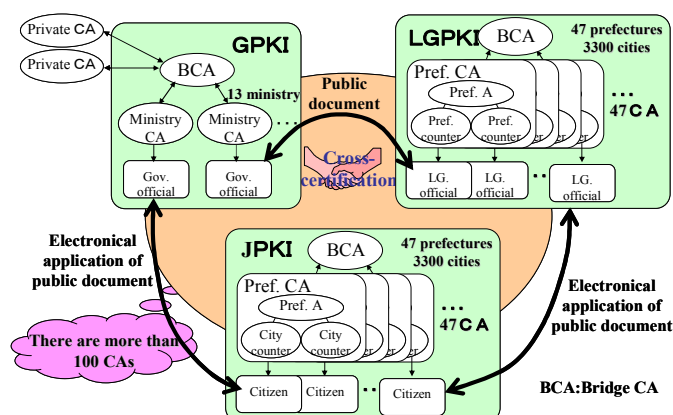


Figure 3. A PKI structure of Japanese public sector.

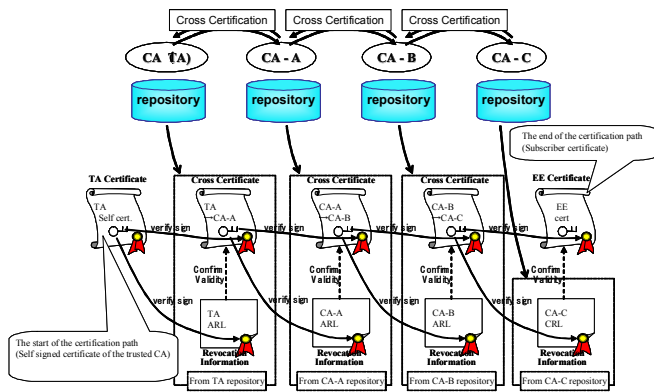


Figure 4. An example of a validation of a certification path.

protocols for certificate validation is Server-Based Certificate Validation Protocol (SCVP) discussed in IETF.

We herein call a server which operates certificate validation processing as CVS (Certificate Validation Server) and made the research on it. When processing on a server, certificate and CRL/ARLs in certification path are verified with public key in certificate. After certificate validation process on a CVS, validation responses are signed by CVS with using CVS private key, and response is verified on client with CVS certificate, as shown in figure 5. In cross-certification model, certification path become long, then CVS has to calculate verify operation with public key many times. An examples as shown in figure 4 certification path length is 4. CVS have 8 times calculations of verify operation of signature value of certificates and CRL/ARL. And because, clients trust a CVS response with verifying signature stored in the response message, then private key of CVS needs to be operated securely. So, CVS need to have much cryptographic computing resource, such as CPU, cryptographic co-processors and HSM (Hardware Security Module) to store private keys securely.

We hereunder report our development of several speed-up methods by using some hardware and software cryptographic modules and the measurement of its effect.

III. EVALUATION OF CRYPTOGRAPHIC OPERATIONS

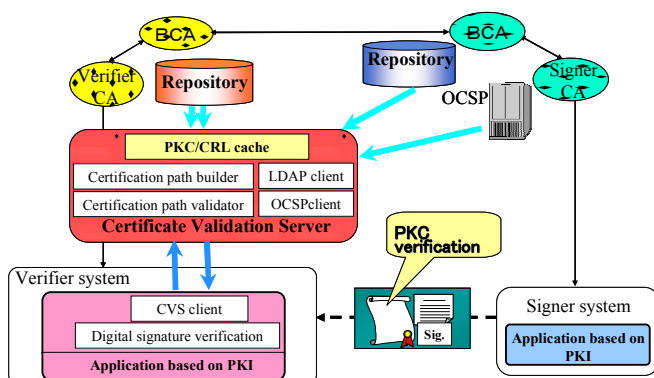


Figure 5. CVS (Certificate Validation Server).

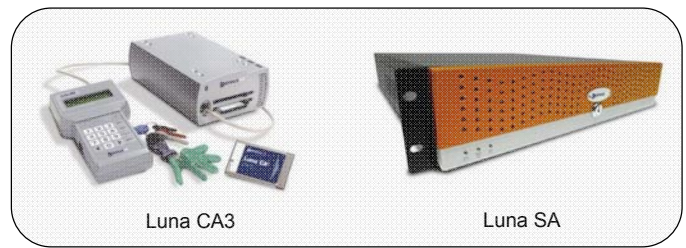


Figure 6. HSM (Hardware Security Module).

The hardware and software cryptographic modules were examined in this paper to speed-up certificate validation function of CVS.

A. Hardware Security Module

To keep private key secure, CVS use HSM for storing private key and signature operation for its response. HSM have a PKCS 11(Cryptographic Token Interface Standard) interface for cryptographic operations. We use two type of HSM, one is a local type, this type is connected the server directly. And another type is network attached type that connects the server with using TCP/IP, as shown in Figure 6. Luna CA3 is a local type HSM and Luna SA is network attached type HSM.

CVS also can use HSM for verifying operations using PKCS 11 interfaces. To verify a signature, at first, we have to make a public key object and store in HSM. And next, we can make verify operation with stored public key object in HSM. Finally we have to delete stored public key object.

B. Software Cryptographic Module

CVS also can use some software cryptographic modules for verifying operation. In this paper we use OpenSSL as software cryptographic module.

C. Generation and Verifying Signatures

In general, generating signature is heavier process than verifying signature. RSA is a most famous public key cryptographic algorithm, and RSA signature operation needs about 20 times processing time than verifying operation in our experiments.

In case of using HSM, the signature operation only needs a pre-stored private key in HSM, and calculates with it. But the verifying operations must store the public key objects into HSM every time whenever the public keys are used, and calculates with them. So, verifying operation will be slower than simple calculation time.

D. Using multiple cryptographic modules

CVS private key must keep secure, the private key store in HSM and signature operation must be done by HSM. So, using multiple cryptographic modules is targeted on verifying operations. When CVS verify signatures of certificate or CRL, the certificate and CRL verifying function in CVS select a cryptographic interface, as shown in Figure 7. The selection of

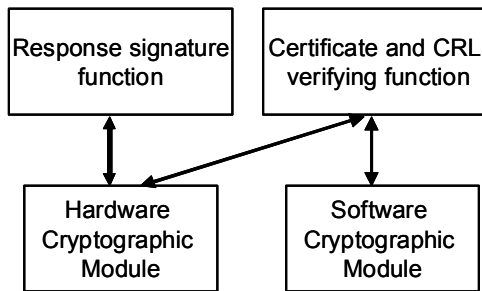


Figure 7. Using multiple cryptographic modules.

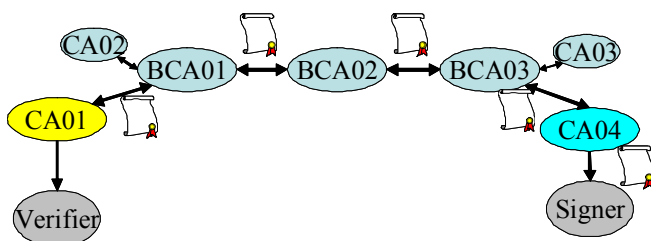
cryptographic modules is configurable, dynamically or statically.

IV. EVALUATE METHOD OF USING MULTIPLE CRYPTOGRAPHIC MODULES

A. Evaluate methods

To evaluate the effect of usage of multiple cryptographic modules for certification path validation processing described in section III, the algorithm using multiple cryptographic modules was integrated into a CVS, and its performance was measured. At first, we measure performances of signature and verifying operation of hardware cryptographic modules and software module respectively. And next, we measure average number of the CVS transactions per unit of time and average response time per unit of time. And next, we measure a CPU and memory usage. We measure these data while changing the numbers of certificate validation requests received from clients simultaneously. We measured 4 cases described under below, and signature operations are always done by HSM.

	Verify by HSM	Verify by software
Luna CA3	Case 1	Case 2
Luna SA	Case 3	Case 4



Source	Destination	Certification Path
CA01 (TA)	Signer	$\text{Cert}_{(CA01 \rightarrow BCA01)}$ $\rightarrow \text{Cert}_{(BCA01 \rightarrow BCA02)}$ $\rightarrow \text{Cert}_{(BCA02 \rightarrow BCA03)}$ $\rightarrow \text{Cert}_{(BCA03 \rightarrow CA04)}$ $\rightarrow \text{Cert}_{(CA04 \rightarrow \text{Signer})}$

Figure 8. An experimental model of certification path.

B. Certification path model

The experimental certification path is shown in figure 8. The path length is 5. CA01 is a Trust Anchor (TA) that issues verifier's certificate. And the target certificate to be validated is signer's certificate issued by CA04. CVS has to verify 5 signatures of certificate and 5 signatures of CRL/ARL.

So, verifying cryptographic operations are 10 times, and signing operation is only once.

C. Experimental environment

The experimental environment is shown in figure 9. The client, the CVS, the LDAP server, and the CA were connected to the same switching HUB. And the network speed was 100Mbps.

The machines used in experimental environment are shown below.

Client

CPU	Pentium 3 1.6GHz
Memory	768MB
Operating System	Windows® XP operating system
HDD	10G

Certificate Validation Server (CVS)

CPU	Ultra SPSTC IIIi 1.28 GHz X 2
Memory	2GB
Operating System	Solaris 8
HDD	80GB
HSM	Luna CA3 (in Case 1 and in Case 2) SA (in Case 3 and in Case 4)

CA and LDAP Directory server

CPU	Pentium 4 3.2GHz
Memory	512MB
Operating System	RedHat Enterprise Linux 4
HDD	80GB

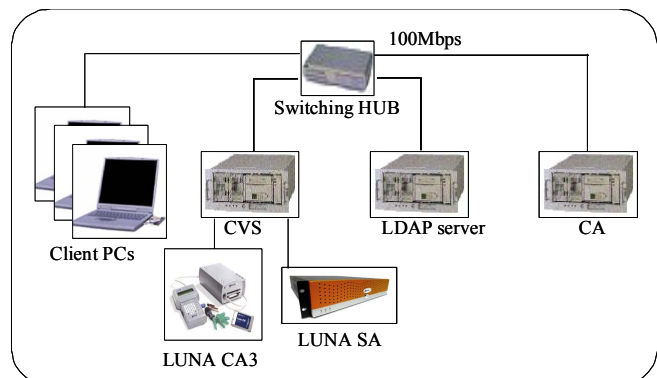


Figure 9. An experimental environment.

V. RESULT AND DISCUSSION

A. Signature and Verifying performances of HSM

Table 1 shows the result of performances of cryptographic modules. In general, verifying operation is faster than signature operation, shown as a result of using OpenSSL for cryptographic module. But, the hardware cryptographic modules have to import public key from the server, before verifying signatures. So, the performance of verifying operation is slower than signature operation.

B. Experimental results and Discussion

Table 2 and Figure 10 show the result of average response time of CVS.

Table 3 and Figure 11 show the result of average throughput of CVS.

Table 4 and Figure 12 show the result of average CPU usage of CVS.

Table 5 and Figure 13 show the result of maximum memory usage of CVS.

Comparing the results of Case 1 and Case 2, also and Case 3 and Case 4 in Figure 10, indicate that load balance of cryptographic operations, in other words, using software cryptographic module make certificate validation processing faster with independence of the number of simultaneous requests.

On the other hand, comparing the results of Case 3 and Case 4 in Figure 10 and 11, indicate that as the number of simultaneous requests increase, the average of response time using hardware cryptographic module was shorter than using software module, and also using HSM increase average of throughput. And In Case 4 CPU usage is higher than in Case 3, regarding a throughput is less than in Case 3 shown in Figure 12 and 13. Using high performance HSM for load balancing of certification path validation process was effective for improving processing speed in overloading.

VI. CONCLUSIONS

The speed-up of certificate validation processing was realized by using multiple cryptographic modules. Especially, with using high performance CPUs for CVS, verifying signatures with software cryptographic modules reduce a response time typically. And it was recognized that the load balance of cryptographic operation is effective for the server executing large amount of validation processing.

Trademarks

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and other countries.

Table 1. SIGNATURE AND VERIFYING PERFORMANCES (TRANSACTION / SEC)

Cryptographic module	Luna CA3	Luna SA	OpenSSL
Signature	25	1260	231
Verify	228	710	4280

Table 2. AVERAGE RESPONSE TIME (SEC/TRANSACTION)

Number of simultaneous requests	1	2	5	10	20	50
Case 1	0.21	0.39	0.94	1.89	3.79	9.50
Case 2	0.08	0.08	0.18	0.37	0.73	1.86
Case 3	0.09	0.11	0.20	0.34	0.63	1.60
Case 4	0.06	0.07	0.17	0.34	0.74	1.90

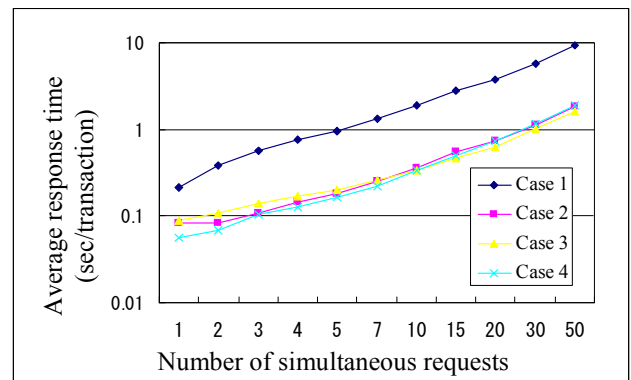


Figure 10. Average response time.

Table 3. AVERAGE OF THROUGHPUT (TRANSACTION / SEC)

Number of simultaneous requests	1	2	5	10	20	50
Case 1	4.6	5.2	5.4	5.4	5.4	5.2
Case 2	12.0	24.0	27.5	27.1	27.3	26.9
Case 3	10.5	18.2	24.6	30.0	29.8	31.0
Case 4	16.8	27.1	28.9	28.4	26.7	26.0

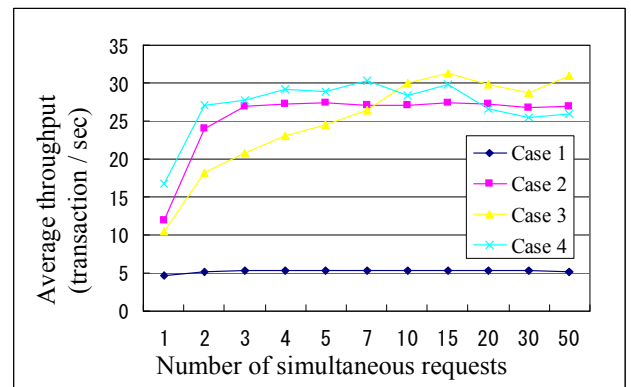


Figure 11. Average of throughput.

Table 4. Average of CPU usage in each simultaneous request (%).

Number of simultaneous requests	1 (%)	2 (%)	5 (%)	10 (%)	20 (%)	50 (%)
Case 1	2.7	3.1	4.5	4.1	4.4	4.8
Case 2	16.8	32.8	57.2	53.0	54.0	55.1
Case 3	13.3	22.2	42.0	49.6	55.0	58.6
Case 4	32.0	49.5	71.9	77.3	78.4	82.3

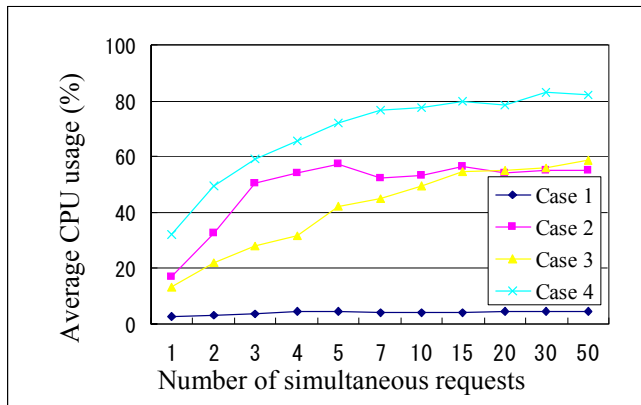


Figure 12. Average of CPU usage.

Table 5. Maximum memory usage (MB).

Number of simultaneous requests	1 (MB)	2 (MB)	5 (MB)	10 (MB)	20 (MB)	50 (MB)
Case 1	32.4	33.4	39.7	46.2	69.3	92.9
Case 2	34.9	36.1	44.5	54.1	73.5	97.1
Case 3	29.6	32.9	46.9	67.7	112.4	172.3
Case 4	62.4	65.0	80.0	101.2	137.9	189.8

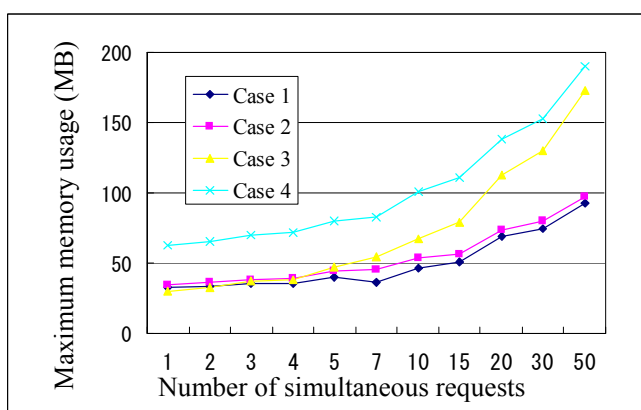


Figure 13. Maximum memory usage.

Intel® and Pentium® are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Sun®, Sun Microsystems® and Solaris®, are registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC™ trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

LUNA CA3 and LUNA SA are trademarks of SafeNet inc. in the United States and other countries.

Red Hat is a trademark of Red Hat, inc. in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States and other countries.

REFERENCES

- [1] ITU-T Recommendation X.509 "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks"(2008)
- [2] A. Frier, P. Karlton and P. Kocher "The SSL 3.0 Protocol" "<http://wp.netscape.com/eng/ssl3/ssl-toc.html>"(1996)
- [3] T. Dierks and C. Allen: RFC 2246:"The TLS Protocol Version 1.0" (1999)
- [4] United State Department of Justice "Electronic Signatures in Global and National Commerce Act", Federal law (USA)
- [5] Japanese Ministry of Internal Affairs and Communications "Law on Electronic Signatures and Certification Services", Law No 102 of 2000 (Japan)
- [6] Chief Information Office Council "Federal Bridge Certification Authority" <http://www.cio.gov/fbca/>
- [7] Japanese Ministry of Internal Affairs and Communications, "Governmental Public key Infrastructure specifications for mutual operation" <http://www.gpki.go.jp/session/CompatibilitySpecifications.pdf> (2001)
- [8] "The Public Certification Service for Individuals" <http://www.jpki.go.jp/>
- [9] T. Fujishiro, T. Kaji, S. Hane, Y. Kumagai and S. Tezuka „Development of Certificate Validation Service” The IEICE Transactions on Information and Systems, PT.1 (Japanese Edition), Vol. J87-D-I No.8, 833 (2004)
- [10] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk: RFC 5280: "Internet X.509 public key infrastructure certificate and Certificate Revocation List (CRL) profile" (2008)
- [11] A. Malpani, S. Galperin, M. Myers, R. Ankney and C. Adams: RFC 2560: "X.509 Internet public key infrastructure online certificate status protocol - ocsp" (1999)
- [12] T. Freeman, R. Housley, A. Malpani, D. Cooper, and W. Polk: RFC5055: "Server-Based Certificate Validation Protocol (SCVP)" (2007)