認証統合および TLS 対応ウェブホスティングサービスの運用

中村 純哉 [1)], 小林 真佐大 [1)], 下條 詠司 [1)], 土屋 雅稔 [1)]

1) 豊橋技術科学大学 情報メディア基盤センター

junya@imc.tut.ac.jp

Operation of Single Sign On and TLS-enabled Web Hosting Service

Junya Nakamura[1)], Masahiro Kobayashi[1)], Eiji Shimojo[1)], Masatoshi Tsuchiya[1)]

1) Information and Media Center, Toyohashi University of Technology

概要

大学の研究活動や業務をサポートするためのウェブサーバは，現代のネットワーク社会で不可欠だが，十分なメンテナンスが行われていないサーバも多く見られる．また，サイバー攻撃が増える中で，サーバメンテナンス技術の高度化も課題となっている．このような背景において，大学の情報系センターが提供するウェブホスティングサービスの役割は増しており，特に認証統合（SSO）と安全な通信（TLS）の実現が重要となっている．SSO は業務効率とセキュリティの向上を目指すもので，TLS は通信のセキュリティを高めるものである．本論文では，豊橋技術科学大学における SSO と TLS に対応したウェブホスティングサービスについて，設計と運用状況を報告する．

1 はじめに

現代のネットワーク社会において，研究成果の広報などの大学における各種業務を円滑に実施するには，安定したウェブサーバが必要不可欠である．しかし，学内組織や研究室に設置されたウェブサーバは，専門外の職員や学生によって維持されている場合が多く，十分にメンテナンスされていないサーバも少なくない．また，クラッキングなどのサイバー攻撃は高度化かつ増加しており，サーバを維持するために必要な技術は高度化する一方である．このような状況を改善するには，各大学の情報系センターがウェブサーバのホスティングサービスを提供することが有効である [19, 22, 26].

近年，2 つの理由から，ウェブホスティングサービスの運用にあたって，認証統合（Single Sign On; SSO）と安全な通信路の確保（Transport Layer Security; TLS）への対応が要請されている．第 1 の理由は，組織内における各種業務情報システムの増加である．これらの情報システムが，個別にユーザ情報を保存し独立にユーザ認証を行うと，利用者の利便性が低下し，安易なパスワードを利用するなどの回避策を採る傾向が強まり，情報システムのセキュリティに対して悪影響を生じる．そのため，情報システムのユーザ情報を統合し，ユーザ認証を連携して，SSO を実現する必要がある [20, 21, 23, 25]．第 2 の理由は，セキュリティ上の要請である．ウェブサーバとクライアントは，Hypertext Transfer Protocol（HTTP）[6] に従って通信を行うが，TLS 化されていない HTTP は，大規模な盗聴や検閲 [10, 17] などの各種の攻撃に対して脆弱である．そのため，Google Chrome や Apple Safari などのブラウザは，2018 年から 2021 年にかけて，TLS 化されていないウェブサーバにアクセスすると警告を出力したり，アクセスを拒否するようになった [3, 8, 9]．これら 2 つの理由から，情報系センターが提供するウェブホスティングサービスにおいても，SSO および TLS に対応する必要がある．

本論文では，豊橋技術科学大学にて構築した，SSO および TLS に対応したウェブホスティングサービスの概要と運用状況について報告する．2023 年 10 月現在，UPKI [24] を含む多くの認証局では 398 日より短い有効期限の TLS サーバ証明書しか発行できない．TLS サーバ証明書は SSO に参加しているシステムの台帳でも必要となることから，サーバ証明書を定期的に更新する作業負荷は無視できない．そこで，ウェブホスティングサービスでは TLS サーバ証明書の更新を自動化することで，この問題に対処する．

本論文の構成は，以下の通りである．2 節では，SSO 参加システム台帳および TLS サーバ証明書の更新を自動化する方法について述べる．3 節では，更新の自動化に対応したウェブホスティングサービスのシステム構成について述べる．4 節で本ホスティングサービ

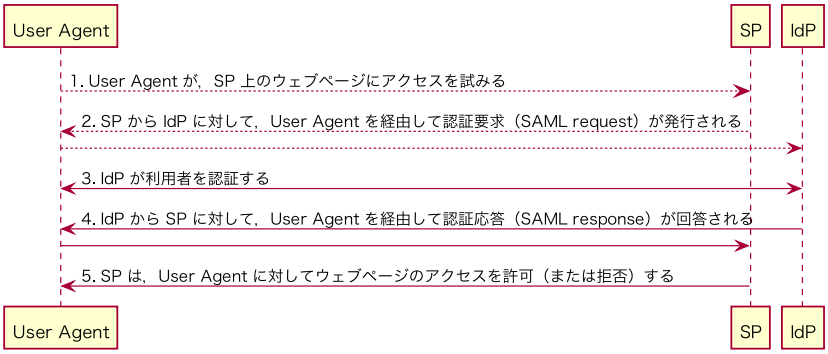

図 1　Shibboleth における認証処理フロー（概要）

スの運用状況について考察し，5 節で結論を述べる．

2　設計

豊橋技術科学大学（以下, 本学）では, 学内ウェブサーバの SSO を実現する基盤システムとして Internet2 Middleware Initiative によって開発された Shibboleth [5] を採用している [20]．**図 1** に，Shibboleth における認証処理フローの概要を示す．本学の Shibboleth は，認証機能と利用者属性情報を提供する Identity Provider（IdP）と，IdP によって認証された利用者に対して実際にサービスを提供する Service Provider（SP）からなり，WAYF サービス（DS）は利用していない．この節では，SSO 参加システム台帳と TLS サーバ証明書の更新を自動化するための設計方針について述べる．

2.1　SSO 参加システム台帳の管理ポリシー

Shibboleth における SSO 参加システム台帳（メタデータ）は，SSO に参加している全ての IdP と SP の公開鍵を，XML 形式で保持している．IdP が SP に対して利用者属性などを格納した認証応答を回答する時（図 1 の第 4 段階），認証応答に含まれる情報が第 3 者によって窃取されないよう，SSO 参加システム台帳に格納されている SP の公開鍵によって暗号化し，IdP の秘密鍵によって署名する．Shibboleth の安全性は SSO 参加システム台帳に依存しているため，SSO 参加システム台帳は適切に管理される必要がある．

Shibboleth の SSO に参加する SP は，HTTPS で用いられる TLS サーバ証明書と，その SP と IdP が通信を暗号化するために用いる証明書（以下，SP 鍵と言う）の 2 つが必要となる．これら 2 つの証明書は，同一のものを用いる必要はない．しかし，別々の証明書を用いる場合，SSO 参加システム台帳に登録する SP 鍵の真性であることを検証することが難しい．IdP 管理者と SP 管理者が対面で SP 鍵を手交することで検証することも考えられるが，遠隔地に SP 管理者がいる場合など，対応困難なケースが存在する．そこで，本論文のウェブホスティングサービスでは，SSO 参加システム台帳に登録する SP 鍵は，既知 SP のホスト名を対象として，UPKI [24] または Let's Encrypt [2] によって発行された TLS サーバ証明書に限る，という管理ポリシーを採用する．

2.2　TLS サーバ証明書更新の自動化

ブラウザのポリシー厳格化 [4, 12, 14] によって TLS サーバ証明書の有効期限が制限されているため，多数のドメインが利用するウェブホスティングサービスでは，TLS サーバ証明書の更新作業を自動化する必要である．そのため，本論文のウェブホスティングサービスでは，一部の特殊ドメインを除いて，証明書発行が自動化されている認証局 Let's Encrypt [2] に証明書の発行を依頼する．

Let's Encrypt による証明書自動発行手順（Automated Certificate Management Environment; ACME）では，証明書の対象となる識別名（ドメイン名等）の所有権を確認する方法が，3 種類ある．ACME クライアントソフトウェア [7] の標準的な方法は，HTTP-01 challenge と呼ばれる手順だが，ACME サーバからウェブサーバに対するネットワーク到達性が必要である．しかし，本学のホスティングサービスにはアクセスを組織内ネットワークに限定しているウェブサーバも含まれるため，HTTP-01 challenge ではなく DNS-01 challenge を採用する．DNS-01 challenge では，ACME サーバから通知されたランダムトークンを特定ドメインの TXT レコードに設定することによって，当該識別名の所有権を証明する．TXT レコードの設定作業は，Dynamic DNS [16] を用いることで自動化した．

2.3　SSO 参加システム台帳更新の自動化

SSO 参加システム台帳および SP 鍵の更新手順は，学術認証フェデレーションによる推奨手順 [1] に準拠

P0) 初期状態
P1) SP は，第 $n-1$ 世代の証明書を使う，かつ，第 n 世代の証明書を復号化のみに使う状態に遷移（SP1）
P2) IdP は，第 n 世代の証明書を SSO 参加システム台帳に追加（IdP2）
P3) SP は，第 n 世代の証明書を使う，かつ，第 $n-1$ 世代の証明書を復号化のみに使う状態に遷移（SP2）
P4) IdP は，有効期限に基づいて，第 $n-1$ 世代の証明書を SSO 参加システム台帳から削除（IdP1）
P5) SP は，第 n 世代の証明書のみを使う状態に遷移，P0 に戻る（SP3）

図 2　証明書更新時の IdP/SP の挙動

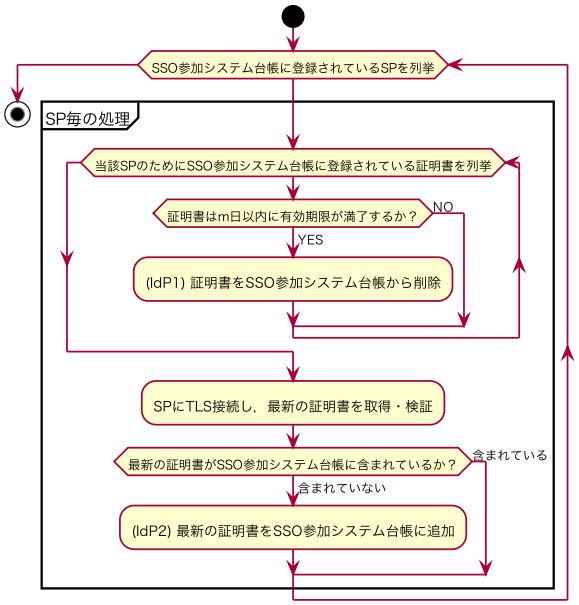

図 3　IdP 上の SSO 参加システム台帳更新手順

する．以降では，SSO 参加システム台帳と SP 鍵を自動的に更新する方法について述べる．

SSO 参加システム台帳および SP 鍵の更新には，少なくとも 2 ノード以上の異なるサーバ（IdP と SP）が関与する．複数のサーバ上でソフトウェアが同時に動作して，同期的に更新処理を進めるという設計は理論的には可能である．しかし，対象となるサーバが非常に多数存在する状況では，更新処理中になんらかのトラブルが発生し，更新処理に失敗する可能性が無視できない．そのため，本システムでは，IdP と SP が独立して非同期に更新処理を行う方針を採る．

SP 上で動作する ACME クライアントソフトウェアが第 n 世代の証明書を取得した時，SSO 参加システム台帳に登録されている第 $n-1$ 世代の証明書を，第 n 世代の証明書に更新する手順を，**図 2** に示す．IdP と SP が独立して非同期に更新処理を行えるよう，図 2 の手順を分解すると，IdP 上の SSO 参加システム台帳更新手順（**図 3**）と SP 上の設定ファイル更新手順（**図 4**）が得られる．SP 上で動作する ACME クライアントソフトウェアが第 n 世代の証明書を取得すると，この最新の証明書は SSO 参加システム台帳には未登録であるから，図 4 の SP1 によって図 2 の P1 に遷移する．次に，IdP から SP に TLS 接続して取得できる証明書が SSO 参加システム台帳に未登録であるから，図 3 の IdP2 によって図 2 の P2 に遷移する．すると，SSO 参加システム台帳には第 n 世代の証明書と第 $n-1$ 世代の証明書の両方が登録されている状態になるため，図 4 の SP2 によって図 2 の P3 に遷移する．第 $n-1$ 世代の証明書の有効期限の残り日数が閾値よりも短くなると，図 3 の IdP1 によって図 2 の P4 に遷移する．すると，SSO 参加システム台帳には第 n 世代の証明書のみが登録されている状態になるため，図 4 の SP3 によって図 2 の P5 に遷移する．以上の動作によって，SP の TLS サーバ証明書と SSO 参加システム台帳の SP 鍵を，安全かつ自動的に更新できる．

3　システム構成

本学のホスティングサービスは，権限管理を担当する認証管理システムと，実際の各種サービスを提供する仮想化基盤システムの 2 種類からなる．このうち認証管理システムについては，[18, 19] で詳細が報告されているため説明は割愛し，ここでは仮想化基盤システムの構成について述べる．ホスティングサービスは 2008 年に運用開始しており，これまでに 2 回システム更新を行っている．ここで紹介するシステム構成は，2021 年より運用している第 3 期システムのものである．

図 5 にホスティングサービスにおける仮想化基盤システムの構成を，**表 1** にシステム仕様を，それぞれ示す．仮想化基盤として，コンテナ型仮想化システムである Docker [11] を採用する．サーバ OS には，Docker の実行に最適化された Container Linux [15] を用いた．各ドメインのサービス稼働状況は，Nagios Core [13] を用いて遠隔監視する．ウェブサイトのデータは，NFS サーバに格納する．

Docker では，コンテナイメージの作成手順（パッケージのインストールや設定ファイルの修正など）を Dockerfile と呼ばれるテキストファイルに記述し，それを Docker に渡すことでコンテナイメージを生成する．本学のウェブホスティングサービスでは，まず，すべてのドメインに共通したコンテナイメージを作成し，このイメージを使って各ドメインのコンテナを実行する．一方で，利用者の要望は一様ではなく，異なる

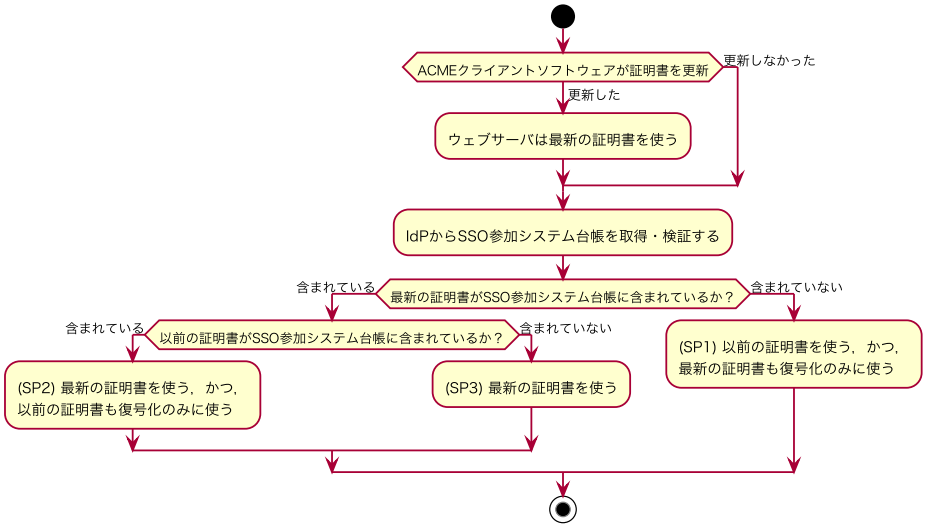

図 4　SP 側の設定ファイル更新手順

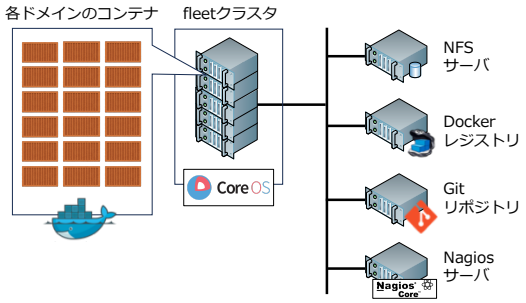

図 5　仮想化基盤システム構成

表 1　システム仕様

コンテナサーバ	
OS	Container Linux 1409.9.0
コンテナランタイム	Docker 1.12.6
ノード仕様	CPU 4 コア，メモリ 12GB
ノード数	5
NFS サーバ	
OS	Debian GNU/Linux 9
CPU	2 コア
メモリ	8GB
ディスク容量	800GB

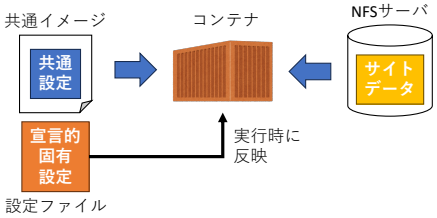

図 6　システムのデータ配置

ソフトウェアスタックを使いたいという要請がある．[*1] ドメインごとに異なる要請に応えるため，ホスティングサービスのコンテナイメージは，ドメインごとの固有設定が宣言的に記述された記述された設定ファイルを実行時に読み込み，反映する仕組みを備える．このように，すべてのドメインで同一の仕様は共通コンテナイメージに集約し，ドメインごとの固有設定は設定ファイルとして管理することで，利用者の要望に応えられる柔軟なサービス提供体制を実現している．最後に，共通イメージを作成するための Dockerfile と，各ドメインの固有設定ファイルは Git でバージョン管理し，変更履歴を追跡できるようにする．

4　運用状況

最初に，ホスティングサービスを開始した 2008 年から 2023 年までの利用ドメイン数の推移を，**図 7** に示す．2008 年度は 34 ドメインの利用申込みがあり，その後 2014 年まで徐々にドメイン数が増加している．それ以後，ドメイン数の伸びは緩やかとなっており，直近 5 年間は 110 ドメイン前後で推移している．学内のウェブサーバは，おおむね移行が完了したものと考えられる．

ホスティングサービスのドメインの中で，TLS を利用しているドメイン数の変化を，**図 8** に示す．本学は 2011 年に UPKI に参加し，ホスティングサービスでも UPKI の TLS サーバ証明書を利用してきた．2022 年から，Let's Encrypt の TLS サーバ証明書への移行を開始した．2023 年 7 月時点において，100 ドメインが Let's Encrypt の TLS サーバ証明書に，移行完了し

[*1] 例えば，ドメイン A は DBMS として MySQL の利用を希望し，ドメイン B では PostgreSQL の利用を希望する場合など．

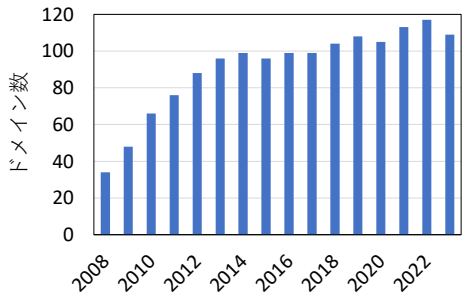

図 7　利用ドメイン数の推移

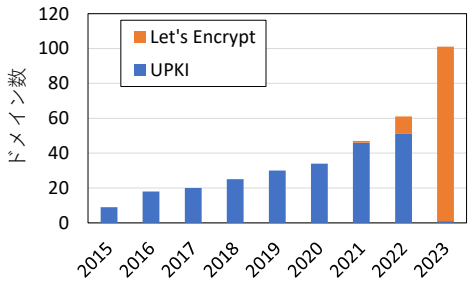

図 8　TLS 利用ドメイン数の変化

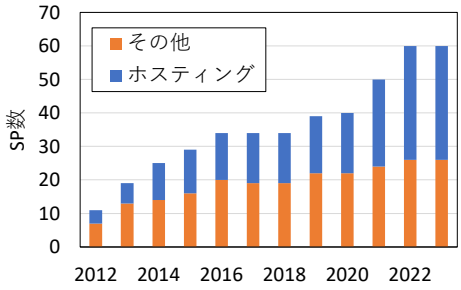

図 9　Shibboleth SP 数の推移

ており，情報メディア基盤センターが管理している特殊ドメイン 1 つのみが UPKI の TLS サーバ証明書を利用している．

本学の SSO に参加している SP 数の推移を，**図 9** に示す．本学において SSO の運用を開始した 2012 年から 2023 年まで一貫して，ホスティングサービスの SP がほぼ半数を占めている．SP を構築することは非経験者にとって非常に困難な作業だが，ホスティングサービスではサービスメニューとして予め用意されているため，各ドメインにおいて Shibboleth によるユーザ認証機能を容易に利用できる．

図 10 に，本学の SSO 参加システムにおける認証回数の推移を示す．先述のように Shibboleth SP のおおよそ半数がホスティングサービスによって構築されているが，認証回数でみると全体の 20% 程度であり，比較的利用頻度の少ないシステムがホスティングサービスでは構築されていることがわかる．

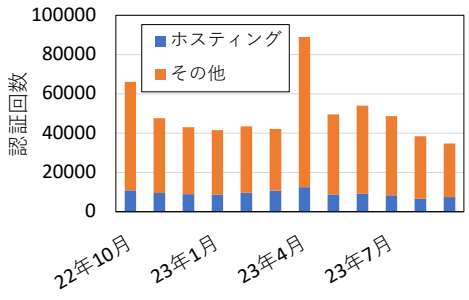

図 10　Shibboleth 認証回数の推移

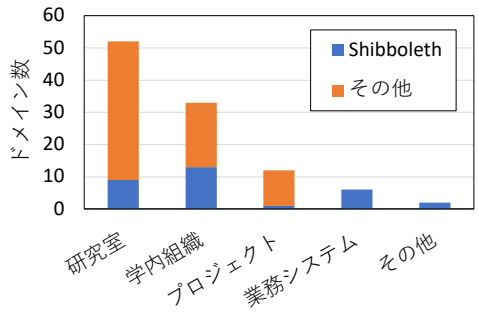

図 11　ホスティングサービス利用ドメインの分類

図 11 に，2023 年 10 月時点におけるホスティングサービス利用ドメインの分類を示す．全体の半数を研究室のウェブサイトが占め，学科やセンターなどの学内組織，各種プロジェクトなどが続く．これらのサイトにおける SSO の利用例としては，研究室所属の教員 · 学生のアカウントに対して個別に研究室専用掲示板のアクセス権限を付与する，事務局ウェブサイトにおいて教員 · 職員というグループに対してアクセス権限を付与するなどがある．

図 12 に，2023 年 6 月の各ドメインのアクセス数の分布を示す．過半数のドメインは，1 ヶ月あたり 5 万アクセス未満である．最もアクセス数の多かったドメインは学内共同施設で，37 万アクセスだった．**図 13** に，アクセス数上位 30 ドメインの HTTP と HTTPS の割合を示す．図から，多くのドメインでは HTTPS への移行が完了していることがわかる．一方で，HTTPS への移行が完了しておらず，引き続き HTTP を主として利用しているサイトも見られる．これは，これらのサイト内のリンクに URL スキームとして `http://` が明示的に書かれていることが原因である．この場合，たとえ HTTPS でサイトを表示しても，サイト内リンクをクリックすると，HTTP のサイトに遷移してしまう．この問題を解決するため，これらのサイトでは既存のウェブページの修正を進めている．

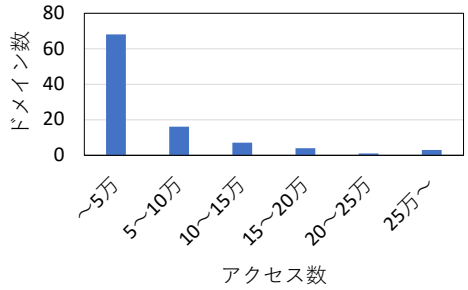

図 12　ドメインごとのアクセス数（2023 年 6 月）

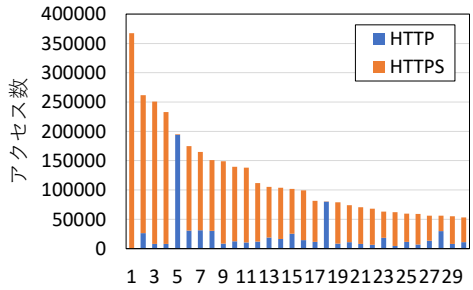

図 13　アクセス数上位 30 ドメインの HTTPS の割合

5　おわりに

本論文では，認証統合（SSO）と安全な通信路の確保（TLS）に対応した大学組織向けウェブホスティングサービスの構築と，その運用状況について報告した．SSO と TLS に対応するには，SSO 参加システム台帳と TLS サーバ証明書の自動更新が必要である．提案システムでは，Let's Encrypt が提供する ACME クライアントを用いて，TLS サーバ証明書の自動更新を実現する．また，認証統合参加システム台帳で用いられる SP 鍵を，ウェブサービスで用いるサーバ証明書と共通化することで，安全かつ効率的に台帳を自動更新する仕組みを構築した．

参考文献

[1] メタデータ記載の証明書更新手順（SP）．https://meatwiki.nii.ac.jp/confluence/display/GakuNinShibInstall/SP+Key+Rollover，（2023 年 10 月 9 日閲覧）．

[2] J. Aas, R. Barnes, et al. Let's Encrypt: An Automated Certificate Authority to Encrypt the Entire Web. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2473–2487, 2019.

[3] Apple. Download macOS Mojave 10.14.4 update, 2019. https://support.apple.com/kb/DL1994，（2023 年 10 月 9 日閲覧）．

[4] Apple．信頼済み証明書に関する今後の制限について，2020. https://support.apple.com/ja-jp/HT211025，（2023 年 10 月 9 日閲覧）．

[5] S. C. (Ed.). *Shibboleth Architecture Protocols and Profiles*, 2005. https://shibboleth.net/documents/internet2-mace-shibboleth-arch-protocols-200509.pdf，（2023 年 10 月 9 日閲覧）．

[6] R. T. Fielding, M. Nottingham, and J. Reschke. HTTP Semantics. RFC 9110, 2022.

[7] E. F. Foundation. Certbot, 2019. https://certbot.eff.org/，（2023 年 10 月 9 日閲覧）．

[8] Google. Chromium Blog: A secure web is here to stay, 2018. https://blog.chromium.org/2018/02/a-secure-web-is-here-to-stay.html，（2023 年 10 月 9 日閲覧）．

[9] Google. Chromium Blog: Increasing HTTPS adoption, 2021. https://blog.chromium.org/2021/07/increasing-https-adoption.html，（2023 年 10 月 9 日閲覧）．

[10] G. King, J. Pan, and M. E. Roberts. How censorship in China allows government criticism but silences collective expression. *American Political Science Review*, 107(2):326—343, 2013.

[11] D. Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, 239(2):2, 2014.

[12] Mozilla. Mozilla Root Store Policy version 2.8.1, 2023. https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/policy/，（2023 年 10 月 9 日閲覧）．

[13] Nagios Enterprises. Nagios Core. https://www.nagios.org/projects/nagios-core/，（2023 年 10 月 9 日閲覧）．

[14] R. Sleevi. Enforce 398-day validity for certificates issued on-or-after 2020–09–01, 2020. https://chromium.googlesource.com/chromium/src/+/ae4d6809912f8171b23f6aa43c6a4e8e627de784，（2023 年 10 月 9 日閲覧）．

[15] The Fedora Project. The container optimized OS. https://fedoraproject.org/coreos/，（2023 年 10 月 9 日閲覧）．

[16] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS UPDATE). RFC 2136, 2000.

[17] X. Xu, Z. M. Mao, and J. A. Halderman. Internet censorship in China: Where does the filtering occur? In N. Spring and G. F. Riley eds., *Passive and Active Measurement*, pp. 133–142, 2011.

[18] 土屋．認証基盤と連携したメールホスティング環境の構築．学術情報処理研究, (13):5–16, 2009.

[19] 土屋．管理者が安全に交代できる学内ホスティングサービス．電子情報通信学会論文誌, J95–B(10):1264–1272, 2012.

[20] 土屋, 中村．事業継続性に配慮した認証基盤システムの構築と運用．情報処理学会論文誌, 63(3):879–894, 2022.

[21] 河野, 稗田, 中村．Shibboleth と OpenAM の連携による認証レベルを制御可能なシングルサインオン基盤の構築．学術情報処理研究, 21(1):71–81, 2017.

[22] 赤尾．京都大学で提供するホスティングサービスの改定．大学 ICT 推進協議会 2013 年度年次大会論文集, 2013.

[23] 浜元, 井田, 齋藤, 小田切, 綿貫, 横山．複数クラウドを利用した 2 段階認証対応全学認証基盤の構築と運用．学術情報処理研究, 24(1):94–103, 2020.

[24] 島岡, 片岡, 谷本, 西村, 山地, 中村, 曽根原, 岡部．大学間連携のための全国共同認証基盤 UPKI のアーキテクチャ設計．電子情報通信学会論文誌, J94–B(10):1246–1260, 2011.

[25] 松平, 笠原, 高田, 東, 二木, 藤田．金沢大学における統合認証基盤の現状と課題．大学 ICT 推進協議会 2013 年度年次大会論文集, 2013.

[26] 平野．Web ホスティングサービス．名古屋大学情報連携基盤センターニュース, 第 3 巻, pp. 7–8, 2004.