情報処理学会論文誌 Vol.59 No.5 1310–1320 (May 2018)

教育用Windows PCアプリケーション実行制御システムにおける複数の証明書チェインを用いた例外的な実行許可

関根 利一[1,†1] 河野 圭太[2] 山井 成良[1,a)] 北川 直哉[1]

受付日 2017年8月17日, 採録日 2018年2月1日

概要：筆者らは大学などの教育機関において学生が利用する教育用 Windows PC の管理方法として，電子証明書の階層構造を用いたアプリケーション実行制御システムを開発した．これにより，改ざんされたアプリケーションの実行を防止したり，複数のアプリケーションをグループ化して一括して実行制御を行ったりすることが可能になった．しかし，これまでの実行制御システムでは，実行禁止グループ内にある特定のアプリケーションを例外的に実行可能にしようとしてもできないなど実行制御の柔軟性に問題があった．そこで本論文では，電子証明書の階層構造において証明書チェインを複数構築することにより，実行禁止グループ内の特定のアプリケーションを例外的に実行許可できる手法を提案する．

キーワード：教育用 PC，アプリケーション実行制御，電子証明書，クロスルート証明書

Exceptional Execution Permission Using Additional Certificate Chains on Application Execution Control System for Educational Windows PCs

Riichi Sekine[1,†1] Keita Kawano[2] Nariyoshi Yamai[1,a)] Naoya Kitagawa[1]

Received: August 17, 2017, Accepted: February 1, 2018

Abstract: In order to manage educational Windows PCs used by students in educational organizations such as universities, we have developed an application execution control system using a hierarchical structure of electronic certificates. This system provides flexible execution control functions such as preventing tampered applications from execution, grouped execution control of multiple applications, and so on. However, this system has some problems in terms of flexibility of execution control such that it cannot allow exceptional execution permission of specific applications in an execution prohibited application group. To solve such problems, in this paper we propose a method enabling exceptional execution permission of prohibited applications by introducing additional certificate chains in a hierarchical structure of electronic certificates.

Keywords: educational PC, application execution control, digital certificate, cross root CA certificate

1. はじめに

大学などの教育機関には，多くの場合，学生が授業や自習で利用できる PC が設置されている．これらの PC は教育用 PC と呼ばれ，一般に学生は大学内のどの PC 端末からでも同一のユーザ環境を利用することができる．これを効率的に実現するために教育用 PC では，学内のサーバで共通のディスクイメージを集中管理し，ネットワークを通してディスクイメージをダウンロードすることで PC を動作させる方法が一般的に利用されている [2], [3]．しかし，教育用 PC を用いた授業では後述するように授業の内容によって，授業ごとに異なる実行させたくないアプリケー

[1] 東京農工大学
Tokyo University of Agriculture and Technology, Koganei, Tokyo 184–8588, Japan
[2] 岡山大学
Okayama University, Okayama 700–8530, Japan
[†1] 現在，株式会社ゼンリンデータコム
Presently with ZENRIN DataCom CO., LTD.
[a)] nyamai@cc.tuat.ac.jp

本論文は文献 [1] を発展させたものである．

ションが存在する．この授業ごとの異なる要求を満たすため，共通ディスクイメージを用いた管理方法では授業ごとに様々なユーザ環境を用意し，提供している．したがって，この方法ではあらかじめシステムの管理者がその数だけイメージディスクを作成して反映させなければならず，この作業は管理者にとって大きな負担となる．

そこで，我々はこの問題の解決のために，教育用 PC 上にあるアプリケーションを実行制御することで個別のユーザ環境を提供するアプリケーション実行制御システムを開発してきた [4]．このシステムでは，教育用 Windows PC 上のアプリケーションに対して個別に実行を許可あるいは禁止と設定することができる．したがって，同じイメージディスクを用いながら，様々な要求に応じたアプリケーション環境を構築することが可能となる．また，ディスクイメージを用いた管理方法とは異なり，授業内の任意の場面で制御を切り替えることも可能である．

加えて，このアプリケーション実行制御をより堅牢に，より柔軟にするために電子証明書を利用する制御方法を考案してきた [5]．この方法ではアプリケーションに電子証明書を用いて署名を行い，その証明書の信頼性を操作することで制御を行う．電子証明書をアプリケーション実行制御に用いる利点には，アプリケーションデータへの改ざん対策と複数のアプリケーションを一括して操作する階層的なグループ制御が可能になることがある．

しかし，従来の制御方法においては下位の証明書が上位の証明書へ持つチェインは 1 つだけであり，実行制御の内容によっては制御ルールの設定が複雑化してしまう問題がある．たとえば，階層的に存在するグループのうち，あるグループに対して実行禁止の制御を行うと，それより下層にあるアプリケーションやグループを例外的に実行許可したい場合でも行うことができず，個別のアプリケーションやグループに対して禁止や許可をそれぞれ設定する必要がある．

そこで問題を解決するため，本論文では複数の証明書チェインを用いた実行制御方法を提案する．この方法では，例外的に実行許可したいアプリケーションやグループに対して，別の証明書チェインを用意する．証明書のチェインが複数作成されることで，アプリケーションやグループは複数のチェインをたどることが可能となる．複数あるチェインのどれか 1 つが有効になっていれば，証明書の信頼性は保証されるため，たとえ禁止グループ下にあったとしてもアプリケーションの実行は許可される．

本論文では，まず 2 章において本論文に関連する電子証明書などの技術について述べる．次に 3 章で，従来の電子証明書を用いたアプリケーション実行制御システムについて述べる．さらに，4 章で，提案手法となる複数の証明書チェインを用いた実行制御について述べる．その後 5 章で，今回の提案手法の動作確認と有効性検証についての実験結果について述べ，6 章で，まとめと今後の課題について述べる．

2. 関連技術

本章では，本論文に関連する技術について述べる．

2.1 電子証明書

電子証明書 [6] とは，様々な電子情報についてその正当性を示すために利用されるものである．この電子証明書は様々な用途に利用され，その利用方法によって SSL 証明書やコードサイニング証明書などと呼ばれる．コードサイニング証明書とは，アプリケーションソフトウェアに対して電子署名を行うために用いられる電子証明書のことを指す [7]．コードサイニング証明書は，それを用いて署名したソフトウェアに対して配布元のなりすましやソフトウェアの改ざんが行われていないことを証明するために用いられる．

2.2 クロスルート方式

クロスルート方式とは，従来の証明書階層構造でのルート証明書に加えて，クロスルート用の中間証明書を設定することにより，別のルート証明書にも接続可能とする仕組みである [8], [9]．この中間証明書のことをクロスルート証明書と呼ぶ．

この仕組みが生まれることになった背景には，古くから電子証明書で用いられている SHA-1 ハッシュ関数の危殆化による，SHA-2 ハッシュ関数への移行推奨がある．これにより，端末にインストールされるルート証明書も最新の電子証明書に更新する必要が発生した．しかし，ルート証明書が更新できずに古いルート証明書を使うことしかできない携帯などの端末では，新しいルート証明書を使用することができない．そこで，これを解決するためにクロスルート方式が考案された．この方式を用いることにより，古いルート証明書しか持たない端末においても，新しいルート証明書を持つ端末においても，各端末が持つルート証明書を用いて検証が可能となり，証明書の信頼性を保証することも可能となる．したがって，このクロスルート方式は複数のルート証明書のうちいずれか 1 つの信頼性が保証されれば，全体としての信頼性も保証される仕組みである．

2.3 複数の署名を持つアプリケーション

市販やフリーのアプリケーションソフトウェアには多くの場合，インストールした段階ではじめから電子署名が付与されている．これらのアプリケーションの電子署名の数は多くの場合で 1 つであるが，中には，複数の電子証明書を用いて署名がされているものも存在する [10]．たとえば，Microsoft 社の Internet Explorer 11 には 2 つの署名がされている．この 2 つの署名の違いは，SHA-1 ハッシュ関数を

用いた署名と SHA-2 ハッシュ関数を用いた署名という点にある．

2.2 節で述べたように，現在では SHA-1 ハッシュ関数ではなく SHA-2 ハッシュ関数の利用が推奨されている．しかし，古い Windows の OS では，SHA-2 ハッシュ関数に対応しておらず，利用することができない．したがって，Internet Explorer 11 などのアプリケーションでは，古い OS でも署名を正しく検証できるように複数の署名が与えられている．どちらかの署名で信頼性を検証することが可能で，その信頼性が保証されれば，ソフトウェアはなりすましや改ざんがなく正規のものであると証明される．

3. 電子証明書を用いたアプリケーション実行制御システム

本章では，我々が従来開発してきた教育用 PC におけるアプリケーション実行制御システム [4], [5] について説明する．まず，システム全体の構成について示し，その後，このアプリケーション実行制御において電子証明書を用いることによる利点と問題点について述べる．

3.1 システム全体の概要

本システムは図 1 で示される 4 つのプログラムで構成されている．それぞれ，教員用の設定ツール，管理者ツール，ポリシデータベース（以下，ポリシ DB）を格納するサーバ上で動作するプログラム，学生用 PC 上で動作する実行制御プログラムとなっている．まず，教員用の設定ツールは，教員が授業を受ける学生に対してどのアプリケーションの実行を許可し，どのアプリケーションの実行を禁止するか設定するためのプログラムである．次に，管理者ツールは管理者が電子証明書を作成し，その電子証明書を用いてアプリケーションに署名を付与するためのプログラムである．また，ポリシ DB ではアプリケーション制御情報を格納しており，制御ポリシが変更されたときには，サーバ上で動作するプログラムがポリシ DB の内容を更新する．最後に，実行制御プログラムは設定されたポリシをもとに教育用 PC 上で実際に制御を行うためのプログラムである．学生が教育用 PC にログインすると実行制御プログラムが起動し，あらかじめポリシ DB に設定されている制御ポリシを取得する．学生が PC を起動している間はこのプログラムはつねに起動している状態となり，教員から制御ポリシの設定通知が送られた場合は，その情報に応じたアプリケーションの実行制御を行う．

教員が設定ツールを用いて教育用 PC にルールを設定するときのシステムの流れを図 2 に示す．まず，教員が設定ツールにログインし，制御ルールの情報を入力する．設定ツールでは，アプリケーション名と制御を行う必要のある教室の IP アドレス範囲を指定する．すると，設定ツールからポリシ DB へ制御ルールの設定要求がされる．ポリシ DB に教員から要求が送られると，サーバ上で動作しているプログラムがポリシ DB 内に格納されている制御情報の書き換えを行い，指定された IP アドレスの PC 内で動作する実行制御プログラムへ新しい制御ルールを送信する．実行制御プログラムは新しい制御ルールを受け取り，学生の利用する PC に反映させる．また，サーバ上で動作するプログラムは実行制御プログラムにルール送信を行った後，教員用設定ツールに制御ルールの設定完了通知を行う．

また，制御ルールの解除は制御ルール設定時と同様に，教員が設定ツールを用いて行い，制御ルールの設定と解除は教員がログインしている間は自由に行うことが可能である．もし，教員が制御ルールを解除せずにログオフした場合，その教員が設定していた制御ルールもその時点で解除される．

学生の利用する PC 上でのアプリケーション実行制御には Windows のグループポリシの機能を使用している．グループポリシの機能では各アプリケーションに対して，起動の許可，禁止を個別に設定することができる．この機能において，アプリケーションを識別するための情報として，実行ファイル名やパス，ハッシュ値，電子証明書（コードサイニング証明書）を用いることができる [11]．このうち，我々のシステムでは電子証明書を利用してアプリケーション実行制御を行っている．電子証明書を用いることによる利点については 3.2 節で述べる．

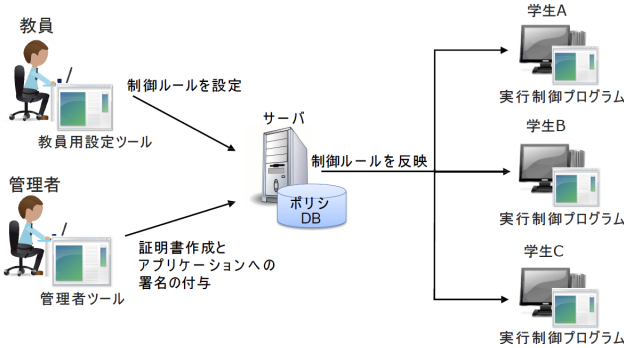


図 1 アプリケーション実行制御システムの構成
Fig. 1 Configuration of application execution control system.

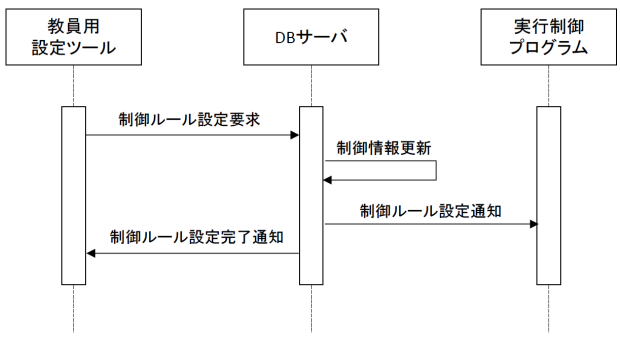


図 2 制御ポリシを設定するときのシステムの流れ
Fig. 2 Operation flow of control rule configuration.

電子証明書によるアプリケーション実行制御では実行を許可する場合，そのアプリケーションに付与された署名に対応した証明書を信頼された証明書として登録する．反対に，その証明書を信頼されない証明書として登録することで，その証明書を用いて署名されたアプリケーションの実行を禁止することができる．実行制御プログラムはこの操作を自動で行うためのプログラムであり，ポリシ DB から受け取った情報をもとに電子証明書の登録操作を行うことでアプリケーションの実行を制御する．

3.2 電子証明書の信頼性操作によるアプリケーション実行制御

電子証明書による実行制御を行うために，学生が使用する教育用 PC 上では Windows のユーザ権限を適切に設定する必要がある．たとえば，もし，学生が証明書の信頼性操作を自由にできた場合，学生はアプリケーションを自由に起動でき，意図した実行制御が不可能となる．したがって，電子証明書の信頼性操作の権限は管理者のみに設定し，学生にその権限を持たせないようにしなければならない．

アプリケーション実行制御において，電子証明書を用いることの利点の 1 つは実行ファイルの改ざんへの対策が可能な点にある．電子証明書は 2 章で述べたように，様々な電子情報に対してその情報が改ざんやなりすましをされることなく，正しいものであるという正当性を証明するものである．たとえば，アプリケーションが改ざんされていないことを示すコードサイニング証明書の正当性判定は次のようにして行われる．アプリケーションを起動する際，まずアプリケーションのデータからハッシュ値を算出する．同時に，アプリケーションの持つ署名データからもハッシュ値を算出する．この 2 つの値を比較し，もし一致していれば，アプリケーションは改ざんされることなく正しいものであると証明される．対して，2 つの値が異なっている場合，アプリケーションの改ざんが行われている可能性がある．したがって，電子証明書を用いることにより，アプリケーションのデータが書き換えられた場合においても，それを判定して，アプリケーションの実行を禁止することが可能となる．

電子証明書を用いることのもう 1 つの利点は，階層的な制御が可能になることである．電子証明書の信頼性は証明書チェインと呼ばれる階層構造で保証がされている．この階層構造の例を**図 3** に示す．図 3 において証明書は大きく 3 種類に分けることができる．階層構造の最上層に位置するものをルート証明書，階層構造の最下層に位置するものをエンド証明書，その間に位置するものを中間証明書と呼ぶ．また，Windows で証明書ファイルのプロパティを開くと，**図 4** で示す証明書チェインが確認できる．図 4 で Root と名前のついた証明書がルート証明書にあたり，Inter1 から Inter3 が中間証明書にあたる．さらにその下の

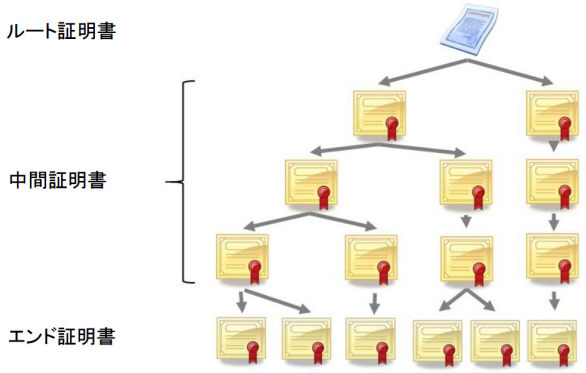


図 3 電子証明書の階層構造
Fig. 3 Hierarchical structure of digital certificates.

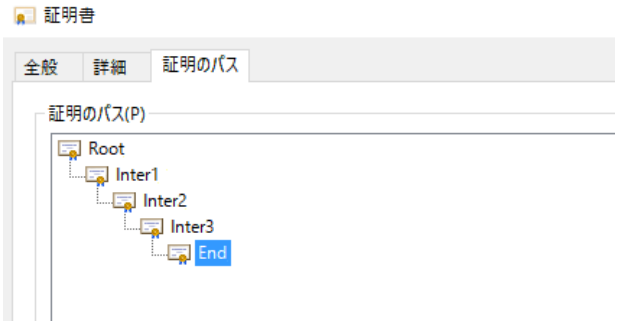


図 4 証明書チェインの例
Fig. 4 Example of certificate chain.

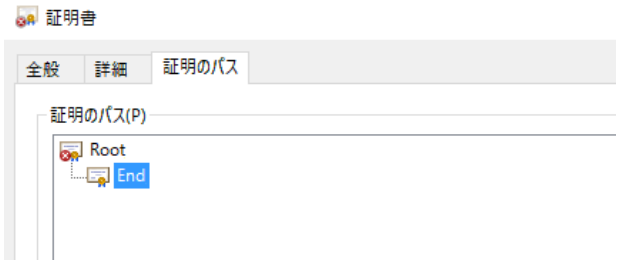


図 5 上位証明書に信頼性が保証されないときの証明書チェイン
Fig. 5 Example of certificate chain with untrusted upper layer certificate.

End がエンド証明書となる．この 3 種類の分類のうち，アプリケーションに付与するコードサイニング証明書は最下層のエンド証明書のことを指す．ルート証明書は下層の中間証明書の信頼性を保証し，中間証明書もまた，下層の中間証明書やエンド証明書の信頼性を保証する．

ある中間証明書やルート証明書の信頼性が保証されなくなった場合，その証明書により信頼性が保証されている下位証明書は保証されず，さらにその下位証明書で保証されている証明書も同時に保証されなくなる．この保証関係が繰り返され，下層にあるすべての証明書の信頼性が保証されない状態となる．上位証明書の信頼性が保証されなくなったときの証明書チェインは**図 5** のように確認できる．図 5 では，Root 証明書の信頼性が保証されていないため，

下位の End 証明書の信頼性もまた保証されていないことが示されている．この信頼性の保障関係を利用し，任意の中間証明書やルート証明書の信頼性を操作することで，その下位にある証明書で署名されたすべてのアプリケーションを一括して操作するグループ制御を行うことが可能となる．

Windows での電子証明書は証明書ストア [12] と呼ばれる場所に格納される．証明書が信頼されるものか，信頼されないものかの判別は，各証明書がストア内のどの場所にインポートされているかにより決定される．証明書ストアのうち，このシステムで用いるストアを**表 1** に示す．証明書が信頼されるものであった場合，ルート証明書は「信頼されたルート証明機関」のストア，中間証明書は「中間証明書機関」のストア，エンド証明書は「信頼された発行元」のストアにそれぞれ格納される．反対に証明書が信頼されないものであった場合は，3 種類の証明書はどれも「信頼されていない証明書」のストアに格納される．それぞれの電子証明書を適切な証明書ストアにインポートすることにより，意図したアプリケーション実行制御を行うことが可能となる．

3.3 従来システムでの問題点

3.2 節で述べたように，アプリケーション実行制御において電子証明書を用いることで堅牢かつ柔軟な制御が可能になる．しかし，従来の証明書階層構造は図 3 のように下位の証明書が上位証明書へ持つチェインは 1 つだけである．したがって，従来のグループ制御方法において設定するルールによっては，グループの一括制御ができずに設定が複雑化してしまう問題がある．

たとえば，証明書階層構造のうち，あるグループを実行禁止するためにその電子証明書を信頼されないものと設定したとする．そのうえで，そのグループよりも下層にあるアプリケーションやグループに対して実行を許可したい場合，従来の方法では証明書信頼性の連鎖により実行の許可をすることができない．これは，ある電子証明書が信頼されないものになると，その証明書よりも下層に存在するすべての証明書はどのストアにインポートをしたとしても，上層の証明書が信頼されないために，信頼された証明書とはならないからである．

もし，実行禁止に設定されているグループ下の単一のアプリケーションを実行許可にしたい場合，証明書ストアの操作による制御では不可能であるため，そのアプリケーションに対して新しいルートからの証明書チェインを付け替える必要がある．しかし，この方法を各アプリケーションに対して行うと，グループ制御が実現しなくなるために従来のシステムでは実装していない．

ゆえに，従来のグループ制御でこの制御を行うには，1 つのアプリケーションやグループの実行許可のために，同じ層にある複数のアプリケーションやグループに対して 1 つずつ実行禁止の設定をしなければならない．加えて，同じ層のアプリケーションやグループの数が多くなればなるほど，この設定には手間がかかる．

また，別の例も説明する．通常時は自由に PC 内にあるアプリケーションを利用できるが，授業 A においては毎回テストを行うことを理由に，テストの解答に関連する情報を得られないようブラウザなど一部のアプリケーションの実行を禁止する必要がある環境を考える．この場合，**図 6** で示すアプリケーションとグループを作成することで環境を構築する．図 6 では，授業 A 中で実行を許可するグループと禁止するグループの 2 つを作成し，各アプリケーションをどちらかに振り分けている．それぞれ App1，App2，App4 は禁止グループ，App3 と App5 は許可グループに属している．授業 A が行われていない通常時はアプリケーションの実行を禁止する必要はないので，どちらのグループも実行を許可するよう両グループ証明書を信頼するストアに格納しておく．そして，授業 A が開始されたときに授業 A 禁止グループの証明書を信頼されないストアに格納することで，授業 A に対応した実行制御が可能となる．

しかし，実際の環境では複数の種類の授業を同じ PC を用いて行っている．したがって，実行制御が必要な授業が図 6 で示した授業 A だけではなく，複数存在する．たとえば，授業 B では授業 A とは異なり，ブラウザの実行は許可するが，計算を行うテストをするために電卓や表計算ソフトなどのアプリケーションを禁止する必要があるかもしれない．ところが，すでにそれぞれのアプリケーションが持つ署名は授業 A 許可グループまたは禁止グループの証明書へのチェインが構築されており，授業 B 許可グループや禁止グループの証明書にチェインをつなげることは不可能

表 1 電子証明書のインポート先
Table 1 Stores to import certificates.

証明書の種類	インポートする証明書ストア	
	信頼	不信頼
ルート証明書	信頼されたルート証明機関	信頼されていない証明書
中間証明書	中間証明書機関	信頼されていない証明書
エンド証明書	信頼された発行元	信頼されていない証明書

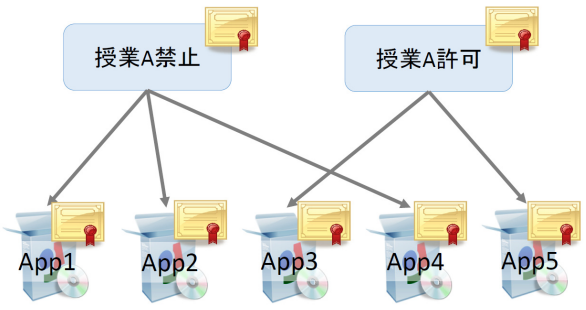


図 6 授業 A に応じたアプリケーションとグループの構成
Fig. 6 Configuration of applications and groups for class A.

である．よって，従来方法ではこのようなグループ化はできず，授業ごとに 1 つ 1 つのアプリケーションに対して個別に実行の許可や禁止を設定する必要がある．

4. 複数チェインを用いたアプリケーション実行制御

3.3 節で述べた問題点を解決するために，本章では電子証明書の階層構造において複数の証明書チェインを構築する方法を提案する．この方法を用いることにより，より柔軟なアプリケーションやグループの制御が可能となる．

4.1 複数チェインを用いたグループ作成

証明書のチェインが複数作成されることにより，アプリケーションやグループは複数のチェインをたどることができる．そのため，従来では 1 つの上位グループにしか属することができなかったアプリケーションやグループが複数の上位グループに属することが可能となる．

本論文で述べる複数チェインの制御は OR にあたる制御である．複数あるチェインのうちの 1 つが無効になったとしても，他のチェインが有効になっていれば証明書の信頼性は保証されるため，アプリケーションの実行は許可される．したがって，アプリケーションの実行を禁止するためには，下位から複数の上位へと接続されるチェインのすべてを無効にする必要があり，すべての上位証明書を信頼されない証明書に設定しなければならない．

複数証明書チェインを構築することで，アプリケーションやグループを複数の上位グループに含めることが可能になるため，3.3 節で問題例として示した実行禁止グループ下にある特定のアプリケーションやグループの例外的な実行許可を行う制御が可能となる．まず，図 3 のような従来どおりの通常のアプリケーショングループ階層構造を作成する．図 3 では上層から下層へのチェインは複数あるが，下層から上層へのチェインは 1 つに限られている．次に，その階層構造内の特定のアプリケーションやグループに対して，グループ禁止適用時でも実行可能にするために上層へのチェインを追加で作成する．詳しいチェインの作成方法については 4.2 節と 4.3 節で述べる．複数あるチェインはどちらか一方が有効になっていれば，下位証明書の信頼性は保証される．したがって，実行が禁止されたグループの下にあるアプリケーションやグループでも，他のチェインが有効になっていれば実行が許可される．

また，3.3 節で述べた各授業に応じたグループ化も可能となる．複数チェインを用いた場合では，たとえば**図 7** で示すアプリケーションとグループを作成し，実行制御を行う．何も授業が行われていない通常時は授業中禁止のグループ証明書を信頼されるストアに格納し，授業 A 許可グループと授業 B 許可グループの証明書を信頼されないストアに格納しておく．授業中禁止のグループはすべてのアプリケーションを含んでいるため，他のグループ証明書が信頼されないものであったとしてもすべてのアプリケーションの実行は許可される．

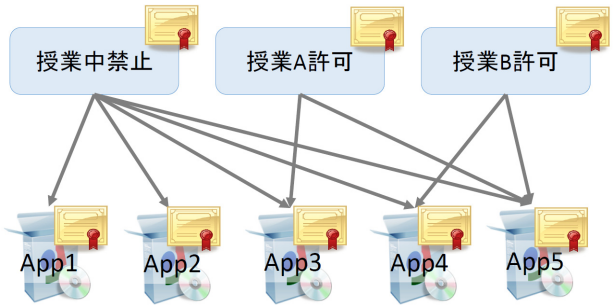


図 7 複数チェインを用いた授業別制御のアプリケーションとグループの構成

Fig. 7 Configuration of applications and groups for multiple classes using multiple certificate chains.

次に，授業 A を行う場合の制御方法を示す．この場合，授業中禁止グループの証明書を信頼されるストアから信頼されないストアに移動させ，授業 A 許可グループの証明書を信頼されないストアから信頼されるストアに移動させる．移動によって，授業中禁止グループと授業 B 許可グループの証明書が信頼されなくなり，App1，App2，App4 の 3 つのアプリケーションは実行が禁止される．対して，授業 A 許可グループに含まれる App3 と App5 はこのグループの証明書が信頼されることにより実行が許可される．また，同様に授業 B を行う場合は授業中禁止グループを信頼されないストア，授業 A 許可グループを信頼されないストア，授業 B 許可グループを信頼されるストアにそれぞれ格納することで，授業 B に応じた制御が可能になる．

従来システムの制御で授業 A と授業 B の要求に応じたアプリケーション実行制御を行う場合は，図 7 のようなグループ化はできないため，始めは許可されていたアプリケーションの電子証明書を 1 つずつ禁止にする必要がある．図 7 の場合では，授業 A と授業 B で禁止するアプリケーション数はどちらも 3 つなので，アプリケーションに対して 1 つずつ実行制御のルール設定をしたとしても，3 つのルール設定で実行制御は可能である．しかし，実際の教育用 PC の環境では非常に多くのアプリケーションが存在する．そして，授業のたびに禁止するアプリケーションの制御ルールについて 1 つずつ設定する必要があるとすると，教員が設定しなければならないルールは複雑となる．対して，図 7 で示したグループ化では，授業中禁止グループの電子証明書とその授業で許可するグループの電子証明書の 2 つを操作するだけで実行制御が可能であり，この数は授業がいくつ増えようとも変わることはない．

複数の証明書チェインを構築するためには 2 つの方法がある．1 つはアプリケーションに複数の署名を付与する方法である．これはアプリケーションに対して行う方法で

あるため，アプリケーションを複数グループに含める場合に用いる．この方法については 4.2 節で詳しく述べる．もう 1 つはクロス証明書と呼ぶ証明書を用いて，証明書を複数の上位証明書にチェインさせる方法である．こちらは中間証明書間で複数チェインを構築する方法のため，グループをグループに含める場合に用いる．この方法については 4.3 節で詳しく述べる．

4.2 アプリケーションへの複数署名付与による制御

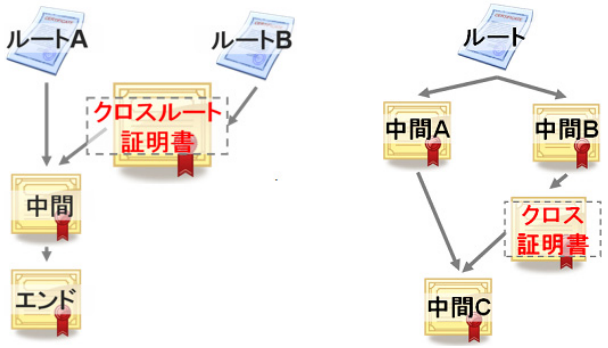
2.3 節で述べたように，市販やフリーのアプリケーションソフトウェアには複数の署名を持つものが存在する．この複数の署名はそれぞれ異なる情報を持つため，別々の署名だと認識される．そして，そのうちの 1 つでも信頼性が保証されれば，そのアプリケーションは信頼できるものであると証明される．アプリケーションに対して複数の署名を付与することにより，1 つのアプリケーションから複数の上位証明書に対してのチェインを構築することができ，それらの信頼性を操作することで実行制御を行うことが可能となる．たとえば，あるアプリケーションが 2 つの証明書で署名されていて，2 つの上位証明書へのチェインを持つ場合，それらの上位証明書のうちの片方の信頼性が保証されなくなり，チェインが無効になったとしても，もう片方の証明書が信頼性が保証され，そのチェインが有効ならばアプリケーションの実行は許可される．両方の証明書の信頼性が保証されず，2 つあるチェインのどちらともが無効になった場合のみ，アプリケーションの実行は禁止される．

ただし，この方法の複数チェインはアプリケーションに対して行うものであるため，階層構造における中間証明書とエンド証明書の間でしか行うことができない．したがって，グループどうしといった中間証明書間での複数チェインの構築には用いることができない．これに対応するために，中間証明書間では 4.3 節で述べるクロス証明書と呼ぶ証明書を用いた制御方法を用いる．

4.3 クロス証明書を用いた制御

一般的な証明書の階層構造において，中間証明書やエンド証明書の上位証明書は 1 つだけである．しかし，電子証明書の階層構造には複数のルート証明書を 1 つの中間証明書につなげるクロスルート方式という構造があり，これは**図 8** (a) の構造で表される．

図 8 (a) のクロスルート証明書はルート A 証明書と同じ秘密鍵と証明書要求ファイル（CSR）[13] を持ちながら，上位証明書をルート B 証明書とした電子証明書である．中間証明書から見ると，ルート A 証明書とクロスルート証明書の中身は同じなので，どちらの証明書も上位証明書と認識される．ゆえに，ルート A 証明書がない環境でもクロスルート証明書とルート B 証明書があれば，中間証明書はクロスルート証明書により，クロスルート証明書はルート B 証明書により署名が検証される．



(a) クロスルート方式
(a) Original method.

(b) クロスルート方式の改良
(b) Modified method.

図 8 クロスルート方式とその改良方式
Fig. 8 Original and modified Cross Root CA certificate.

しかし，アプリケーション実行制御システムでは通常の証明書階層構造において 1 枚しかないルート証明書の信頼性を操作することにより，すべてのアプリケーションを一括で制御できる機能が必要となる．もしもルート証明書が複数存在した場合，最上位の証明書が複数存在することになり，1 枚の証明書のみですべてのアプリケーションの実行を禁止する制御は不可能である．

そこで，このクロスルート方式を改良し，図 8 (b) で表される構造の証明書チェインを作成する．図 8 (b) におけるクロス証明書は，中間 A 証明書と同じ秘密鍵と証明書要求ファイルを用いて，中間 B 証明書により署名がされている．

この証明書構造を用いることで，単一の証明書から複数の上位証明書へのチェインを構築し，アプリケーショングループをその上位の複数グループに含ませることができる．図 8 (b) において，中間 A 証明書の信頼性が保証されなくなった場合でも，クロス証明書と中間 B 証明書の信頼性が保証されていれば，中間 C 証明書の信頼性は保証される．したがって，この方法も 4.2 節で述べたアプリケーションに複数の署名を付与する方法と同様に，複数のグループに含まれているアプリケーションやグループは，複数ある上位グループのすべてに禁止ルールが設定されていない限り，その実行は許可される．

また，図 8 (b) の複数証明書チェインにおいて，どちらのチェインの信頼性も保証されている場合，信頼性の検証はより階層数が少ない左側のチェインにて行われる．しかし，中間 A 証明書が信頼されない場合は右側のチェインにて証明書検証が行われるため，階層数が 1 つ増加する．したがって，4.2 節で述べた複数署名の方法とは異なり，証明書検証を行うチェインによっては階層数が増加する可能性がある．階層数が増加するにつれて検証にかかる証明書数や時間は増加するため，階層数は少ないほうがよい．以上より，4.2 節の制御方法が利用できる中間証明書とエン

ド証明書間では，こちらの方法を利用する．

4.4 複数チェインを用いたシステムの実装

アプリケーション実行制御システムにおいて 4.2 節と 4.3 節で述べた複数チェインを扱うための実装について説明する．実行制御システムは従来システムと同様に，教員用設定ツール，管理者ツール，ポリシ DB を格納するサーバ上で動作するプログラム，実行制御プログラムで構成されるが，それぞれのプログラムで複数チェインを持つ証明書構造が利用できるように変更を行った．

アプリケーションに署名を付けるためには，従来システムと同様に Windows ソフトウェア開発キットである Windows SDK に含まれる signtool を用いたが，この際にコマンドオプションの 1 つである /as オプションを指定することで，アプリケーションにさらに電子署名を追加することが可能となる．そこで，管理者ツールの機能を拡張し，アプリケーションに複数署名を付与できるようにした．また，クロス証明書の作成も管理者ツールから可能になるように機能を追加した．

ポリシ DB は制御情報の管理に加え，アプリケーションと電子証明書の紐付けや証明書チェインの管理などを行っているため，アプリケーションに付けられた複数の署名やクロス証明書についても登録と管理が可能になるようサーバ上で動作するプログラムも含めて改良した．さらに，実行制御プログラム内で行われる電子証明書のストア操作においても，アプリケーションに付けられた複数の署名に対応した電子証明書やクロス証明書が証明書ストアにインポートされるように改良を行った．

5. 実験

本章では，4 章で述べた制御方法が実際に動作することの確認，および有効性を検証するための実験結果について述べる．

5.1 動作確認実験

4 章で提案したアプリケーションに複数の署名を付与する方法と複数の上位証明書を用いる方法のそれぞれについて，アプリケーション実行制御システムを用いて意図した制御が行われるか確認するための実験を行った．証明書階層構造を作成し，それぞれの証明書をルール変更により信頼されない証明書としたとき，署名されたアプリケーションがどのように動作するか確認した．

本実験を行うために作成した証明書チェインと署名を付与したアプリケーションの構成を図 9 で示す．End1 と End2 の証明書はエンド証明書，InterA から InterD までの証明書は中間証明書，Cross 証明書はクロス証明書を表している．

test.exe には，End1 証明書と End2 証明書の 2 つの証明書を用いて署名を付与している．End1 証明書は InterA 証明書により署名されており，InterA 証明書は Root 証明書により署名されている．また，End2 証明書は InterD 証明書により署名されており，InterD 証明書は InterB 証明書により署名されている．さらに，Cross 証明書は InterB 証明書と同じ秘密鍵と証明書要求ファイルで作成され，InterC 証明書で署名されている．最後に，InterB 証明書と InterC 証明書は Root 証明書により署名されている．また，これらすべての証明書は信頼されるストアに格納しておく．その上で，Windows のグループポリシの証明書の規則を適用し，初期状態において test.exe の実行を許可する．以下に実験の手順を示す．

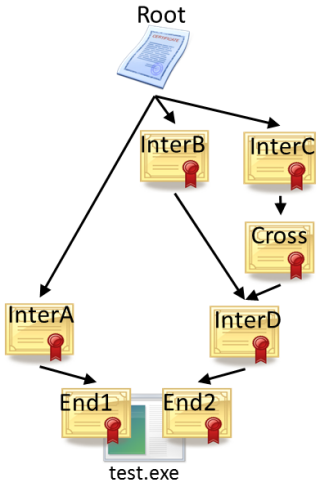


図 9 複数証明書チェインによる制御実験の証明書階層構造
Fig. 9 Example of certificate hierarchical structure of control experiment with multiple certificate chains.

(1) 実行制御プログラムを起動する．
(2) 設定ツールを用いて InterD グループの禁止制御を行う．
(3) test.exe が実行できることを確認する．
(4) 設定した InterD グループの制御ルールを削除する．
(5) 設定ツールを用いて InterA グループの禁止制御を行う．
(6) 証明書チェインがどのように行われているか確認し，test.exe が実行できることを確認する．
(7) 設定ツールを用いて InterB グループの禁止制御を行う．
(8) 証明書チェインがどのように行われているか確認し，test.exe が実行できることを確認する．
(9) 設定ツールを用いて InterC グループの禁止制御を行う．
(10) test.exe の実行が禁止されていることを確認する．

上記の手順に従って，実験を行った結果を示す．まず，実行制御プログラムを起動し，教員用設定ツールから InterD グループの実行禁止ルールを設定した結果，InterD 証明書が信頼されない証明書のストアに格納されたことが確認で

きた．さらに，test.exe の実行が許可されているか確認を行ったところ，問題なく実行することができた．また，このときの test.exe の署名情報を**図 10** で示す．図 10 の左側は End1 証明書の詳細であり，その信頼性は保証されていた．対して，右側は End2 証明書の詳細であり，こちらの証明書の信頼性は保証されていなかった．

次に，InterD グループに対して設定していた実行禁止ルールを削除し，InterA グループの実行禁止ルールを設定した結果，信頼されない証明書のストアでは InterD 証明書が削除され，InterA 証明書がインポートされていた．この場合でも，test.exe が実行できることを確認した．また，test.exe の署名を確認すると，InterD グループ禁止時とは反対に End1 証明書の信頼性は保証されておらず，End2 証明書の信頼性は保証されていた．このときの End2 証明書から Root 証明書へのチェインを**図 11** で示す．InterD 証明書は InterB 証明書により署名を検証され，InterB 証明書は Root 証明書により署名を検証されていた．

さらに，InterB グループに対して実行禁止ルールを設定した結果，信頼されない証明書のストアに InterB 証明書がインポートされた．test.exe が実行できるか確認したところ，この場合でも test.exe の実行は許可されていた．このときの End2 証明書から Root 証明書へのチェインを**図 12** に示す．InterD 証明書は InterB 証明書により署名を検証され，さらに InterB 証明書は InterC 証明書により署名が検証されていた．ただし，この InterB 証明書は InterB 証明書と同じ秘密鍵と証明書要求ファイルで作られた Cross 証明書である．したがって，InterD 証明書は Cross 証明書，InterC 証明書をたどって Root 証明書へとチェインされていた．最後に，InterC グループに対して実行禁止ルールを設定すると，test.exe の実行が禁止されていた．

これらの結果から，アプリケーションに複数の署名を付与する方法と複数の上位証明書を用いる方法のどちらにおいても，アプリケーション実行制御システムでのルール設定から実行制御まで想定した動作が行われることを確認できた．

図 10 InterD グループ禁止制御時における test.exe の署名情報
Fig. 10 Signature information of test.exe on execution prohibition.

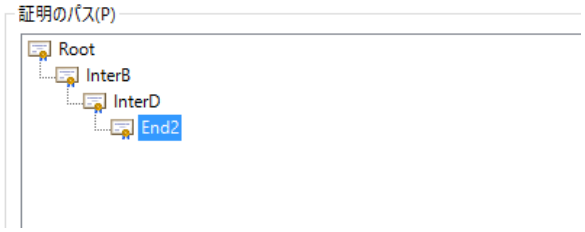


図 11 End2 証明書から Root 証明書へのチェイン
Fig. 11 Certificate chain from End 2 certificate to Root certificate.

5.2 アプリケーション起動時間の計測

アプリケーション実行制御システムでは，アプリケーションの種類やバージョンによって階層的にグループが作成されている．そのため，クロス証明書を含んだ証明書チェインによるアプリケーション実行制御において，階層数の増加がアプリケーション起動までの時間にどの程度影響を与えるか確認する必要がある．この実験では，クロス証明書を含む証明書チェインの階層数を数段階に変え，それぞれの階層数におけるアプリケーション起動までの時間を計測した．時間計測には Windows PowerShell の Measure-Command [14] を用い，実行ファイルは起動後即終了するだけのものとした．この実験により，アプリケーション実行制御において電子証明書を利用し，複数チェイン構造を用いることがシステムの運用の妨げにならないことを示す．

実験のために，同じ内容の exe 実行ファイルを複数作成した．それぞれの実行ファイルは，電子署名を持たないもの，ルート証明書とエンド証明書の 2 階層の階層構造を持つ証明書で署名されたもの，ルート証明書からエンド証明書まで 5 階層の階層構造を持つ証明書で署名されたもの，ルート証明書からエンド証明書まで 10 階層の階層構造を持つ証明書で署名されたものとした．以下の手順に従い，実験を行った．

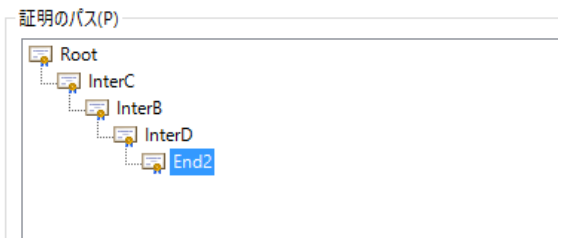


図 12 InterB グループ禁止制御時の End2 証明書から Root 証明書へのチェイン
Fig. 12 Certificate chain from End 2 certificate to Root certificate on execution prohibition of InterB group.

(1) Windows OS を再起動させる．
(2) Windows PowerShell を起動させる．
(3) Measure-Command を用いて，電子署名をもたない実行ファイルの起動時間について 5 回計測する．
(4) Measure-Command を用いて，2 階層の証明書により署名された実行ファイルの起動時間について 5 回計測する．
(5) Measure-Command を用いて，5 階層の証明書により署名された実行ファイルの起動時間について 5 回計測する．
(6) Measure-Command を用いて，10 階層の証明書により署名された実行ファイルの起動時間について 5 回計測する．

実験の計測結果を**表 2** に示す．証明書なしの場合と証明書を付け加えた場合のどちらにおいても，1 回目の起動時間よりも 2 回目以降の起動時間が速くなった．これは，アプリケーションの起動情報を PC 内部でキャッシュしていることによるものだと考えられる．対して，すべての実行ファイルの 1 回目の起動は，Windows OS を再起動した後の 1 回目起動であるため，キャッシュは行われていない．そのうえで，すべての実行ファイルの 1 回目の起動時間は約 4.8〜5.6 ms となっている．この結果から，Windows OS を起動して 1 回目のキャッシュがされていない状態の起動でも，起動時間は十分短いといえる．

また，証明書なしの場合と比べて証明書を付け加えた場合は 1 回目の起動平均時間が約 0.3〜0.5 ms，2 回目以降の起動平均時間が約 0.5〜0.6 ms 遅くなっている．この遅延時間がアプリケーション実行制御で証明書検証を利用していることによる影響だと考えられる．なお，証明書有無による起動時間の違いはあったが，証明書チェインが 2 階層，5 階層，10 階層と階層が増えたことと平均起動時間の間には関連性が見つけられなかった．本来は階層が増えれば，その分だけ証明書検証のために時間を要するはずであるが，このようになった理由は不明である．

我々のシステムで用いる証明書階層構造は 10 階層程度を想定しており，この実験結果から証明書の検証がアプリケーションの起動時間に与える影響は十分小さいものだと判断される．したがって，システムの運用にあたり，起動時間の問題は妨げにならないことが確認できた．

表 2 起動時間の計測結果
Table 2 Result of start-up time measurement.

	起動時間［ms］			
	署名なし	2 階層	5 階層	10 階層
1 回目	4.8894	5.6647	5.6240	5.1527
2 回目	1.5659	3.5871	3.6463	3.5196
3 回目	1.0899	1.4562	1.1093	1.2261
4 回目	1.1057	1.0828	1.1140	1.1076
5 回目	1.0757	1.1033	1.1136	1.0907
平均（2〜5 回目）	1.2093	1.8073	1.7458	1.7630

6. おわりに

本論文では，教育用 PC におけるアプリケーション実行制御において証明書の複数チェインを利用した方法について述べた．これまでのアプリケーション実行制御では，下位の証明書が上位の証明書へ持つチェインは 1 つだけであり，実行制御の内容によっては制御ルールの設定が複雑化してしまう問題があった．この解決のために，証明書階層構造において複数チェインを構築する方法を提案し，アプリケーションへの複数署名の方法とクロス証明書を作成する方法の 2 通りを説明した．加えて，動作確認と有効性の検証を行い，システムの運用上の問題はないことを確認した．

今後の課題として，より柔軟な制御を可能にするため，複数ある上位証明書のすべてが信頼されているときのみ実行が許可される制御方法の検討があげられる．今回の複数証明書チェインでは，複数ある上位証明書のうち 1 つでも信頼されていれば実行が許可された．この場合，実行を禁止するためにはすべての上位証明書の信頼性を無効にする必要がある．これに対し，複数ある上位証明書のすべてが信頼されているときのみ実行が許可される制御方法の場合，上位証明書のうちの 1 つの証明書の信頼性を無効にするだけで実行の禁止ができる．

たとえば，インストールされているすべてのブラウザが入るブラウザグループと，インストールされているすべての Microsoft 社製品が入る Microsoft グループが教育用 PC 内で設定されており，Internet Explorer 11 アプリケーションは両方のグループに属しているとする．もし，教員が授業でブラウザ全体を禁止したい場合，ブラウザグループを禁止設定することで制御を行えることが望ましい．しかし，本論文で述べた証明書チェインは複数あるチェインのうちの 1 つが無効になったとしても，他のチェインが有効になっていれば証明書の信頼性は保証されるため，ブラウザグループを禁止設定しただけでは Microsoft グループにも属する Internet Explorer 11 アプリケーションの実行は禁止されない．もし，複数ある上位証明書のすべてが信頼されているときのみ実行が許可される制御方法が利用できれば，アプリケーションの属するグループのうちの 1 つを禁止設定にするだけでそのアプリケーションを実行禁止にすることが可能となるため，ブラウザグループの禁止設定で Internet Explorer 11 を実行禁止することが可能になる．

参考文献

[1] 関根利一，岡本大輔，山井成良，北川直哉，河野圭太：教育用 PC における電子証明書の信頼性操作と複数の証明書チェインによる柔軟なアプリケーション実行制

御，情報処理学会研究報告インターネットと運用技術，Vol.2016-IOT-33, No.1, pp.1–6 (2016).
[2] 上田 浩，喜多 一，森 幹彦，石井良和，外村孝一郎，植木 徹，上原哲太郎，梶田将司：ネットブートとデスクトップ仮想化を採用した京都大学の教育用端末系の構築：TCO 削減を目指して，インターネットと運用技術シンポジウム 2012 論文集，pp.47–54 (2012).
[3] 奥村 勝，藤村 丞：1000 台規模のディスクレス PC システムの構築と運用，情報処理学会研究報告インターネットと運用技術，Vol.2008, No.15, pp.61–66 (2008).
[4] Kawano, K., Okamoto, D., Fujiwara, M. and Yamai, N.: A Flexible Execution Control Method of Application Software for Educational Windows PCs, *Journal of Information Processing*, Vol.22, No.2, pp.161–174 (2014).
[5] Okamoto, D., Kawano, K., Yamai, N. and Yokohira, T.: Strict Application Execution Control with Hierarchical Group Management Using Digital Certificates on Educational Windows PCs, *Journal of Information Processing*, Vol.23, No.4, pp.449–457 (2015).
[6] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. and Polk, W.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC5280 (2008).
[7] Symantec：Microsoft Authenticode 用コードサイニング証明書（オンライン），入手先 ⟨http://www.symantec.com/ja/jp/code-signing/microsoft-authenticode⟩（参照 2017-06-10）.
[8] Global Sign：[EV SSL] クロスルートとは何ですか（オンライン），入手先 ⟨https://jp.globalsign.com/support/faq/431.html⟩（参照 2017-06-10）.
[9] Symantec：クロスルート設定用証明書の設定について，どのような対応が必要でしょうか（オンライン），入手先 ⟨https://knowledge.symantec.com/jp/support/ssl-certificates-support/index?&page=content&id=SO28069⟩（参照 2017-06-10）.
[10] Symantec：Microsoft Authenticode - Dual Code Signing Instructions with SHA1 & SHA256 hashing Algorithm (online), available from ⟨https://knowledge.symantec.com/support/code-signing-support/index?page=content&id=INFO2274⟩ (accessed 2017-06-10).
[11] Microsoft Developer Network：ソフトウェア制限ポリシーの規則を使用（オンライン），入手先 ⟨https://msdn.microsoft.com/ja-jp/library/hh994597.aspx⟩（参照 2017-06-10）.
[12] Microsoft TechNet：証明書ストアを表示する（オンライン），入手先 ⟨https://technet.microsoft.com/ja-jp/library/cc725751.aspx⟩（参照 2017-06-10）.
[13] Nystrom, M. and Kaliski, B.: PKCS #10: Certification Request Syntax Specification Version 1.7, RFC2986 (2000).
[14] Microsoft TechNet：Widows PowerSheall の機能 - Measure-Command コマンドレットの使用（オンライン），入手先 ⟨https://technet.microsoft.com/ja-jp/library/ee176899.aspx⟩（参照 2017-08-06）.

関根 利一 （正会員）

平成 28 年東京農工大学工学部情報工学科卒業．平成 30 年同大学大学院工学府情報工学専攻博士前期課程修了．在学中，分散システム，インターネットアーキテクチャの研究に従事．

河野 圭太 （正会員）

平成 12 年大阪大学工学部電子情報エネルギー工学科卒業．平成 14 年同大学大学院工学研究科博士前期課程修了．平成 16 年同大学大学院情報科学研究科博士後期課程を修了し，同年岡山大学総合情報基盤センター助手．平成 19 年同センター助教，平成 22 年同大学情報統括センター助教を経て，平成 23 年同センター准教授．博士（情報科学）．モバイルネットワーク，分散システムの研究に従事．IEEE，電子情報通信学会各会員．

山井 成良 （正会員）

昭和 59 年大阪大学工学部電子工学科卒業．昭和 61 年同大学大学院博士前期課程修了．昭和 63 年同大学大学院基礎工学研究科（物理系専攻情報工学分野）博士後期課程退学．同年奈良工業高等専門学校情報工学科助手．同講師，大阪大学情報処理教育センター助手，同大学大型計算機センター講師，岡山大学総合情報処理センター（現，情報統括センター）助教授を経て，平成 18 年同教授．平成 26 年より東京農工大学大学院工学研究院教授．分散システム，ネットワーク運用管理，ネットワークセキュリティの研究に従事．IEEE，電子情報通信学会各会員．博士（工学）．本会シニア会員．

北川 直哉 （正会員）

平成 21 年中京大学情報科学部情報科学科卒業．平成 23 年同大学院博士前期課程修了．平成 26 年名古屋大学大学院情報科学研究科情報システム学専攻修了．同年東京農工大学大学院工学研究院先端情報科学部門助教．ネットワークセキュリティ，情報セキュリティの研究に従事．博士（情報科学）．