2005－OS－99（7）
2005／5／26

携帯電話向けアクセス制御機能付きX Server

朝倉 義晴 †，本田 篤 †，稗田 諭士 †，千嶋 博 †

† ＮＥＣ システムプラットフォーム研究所

携帯電話が提供するサービスの増加にともない，携帯電話のセキュリティ強化の要望はますます高まっている．本稿は，安全なネイティブアプリケーション追加サービスの実現を目的とした携帯電話のアプリケーション実行環境のセキュリティ強化の取り組みとして，携帯電話向けのアクセス制御機能付きX Serverを提案する．アクセス制御機能付きX Serverは，X Clientから受信したXリクエストの実行を制御することで，X ClientによるXリソースに対するアクセス制御を行う．さらに，アプリケーション実行環境が携帯電話として振る舞うために，アプリケーション実行環境に求められる要求を考慮したアクセス制御も行う．X Serverにアクセス制御機能を追加することで，悪意あるX Clientによる他のX Clientの保有するXリソースに対する不正なアクセスを防げる．

Access Control X Server for Mobile Phones

Yoshiharu ASAKURA †　Atsushi HONDA †　Satoshi HIEDA †　Hiroshi CHISHIMA †

† System Platforms Research Laboratories, NEC Corporation

Recently mobile phones have required security functions to provide new services for users securely. This paper proposes the access control X Server for mobile phones to realize a secure execution of downloadable native applications. The access control X Server realizes an access control by means of controlling execution of X requests, which are sent by X Clients. Furthermore, the access control X Server satisfies requirements of an application execution environment for mobile phones. This access control function protects X resources against access of malicious X Clients.

1　はじめに

近年，携帯電話は無線LANやBluetooth [1] などの外部接続手段を備えるようになり，携帯電話にインストール可能なコンテンツの種類も増加している．このような携帯電話の高機能化にともない，携帯電話が提供可能なサービスの種類も増加している．例えば，携帯電話へインストール可能なコンテンツは，当初はメロディや画像などのデータのみであったが，現在はアプリケーションをインストールして実行することも可能である．携帯電話向けサービスの発展にともない，携帯電話へ外部からデータやアプリケーションをインストールする機会はますます増加すると考えられる．このような状況の中で，携帯電話のセキュリティに対する注目が高まっている．携帯電話のセキュリティ上の懸念としては，Bluetoothなどの外部接続手段を経由した携帯電話に対する不正アクセスやウィルスなどの悪意あるアプリケーションの実行などが挙げられる．

携帯電話に期待されているサービスとして，ネイティブアプリケーション追加による携帯電話の機能拡張サービスがある．また，ネイティブアプリケーションの追加が可能な携帯電話のアプリケーション実行環境もすでに存在する [1) 2)]．ネイティブアプリケーションの追加が可能になることで，携帯電話の機能拡張の幅が広がり，より多様な用途に携帯電話を利用できる．しかし，ネイティブアプリケーションは携帯電話のリソースに直接アクセスできるため，悪意やバグを原因とした携帯電話が提供するサービスの妨害や個人情報漏洩の危険性が高まってしまう．筆者らのグループは，安全なネイティブアプリケーション追加サービスを実現するために，携帯電話のアプリケーション実行環境のセキュリティ強化に取り組んでいる．セキュリティ強化の取り組みとして，アクセス制御とリソース制御がある．アクセス制御は，ネイティブアプリケーションによる携帯電話のリソースへ

[1] Bluetooth は Bluetooth SIG, Inc. の商標である．

のアクセスを制御する．リソース制御は，ネイティブアプリケーションによる携帯電話のリソース使用量を制御する．筆者らの対象としているアプリケーション実行環境は，OS として Linux [2] を使用し，ウィンドウシステムとして X Window System [3] を使用する．よって，GUI を使用するネイティブアプリケーションは X Client である．

本稿では，携帯電話のアプリケーション実行環境のアクセス制御強化として，X Client による X リソースへのアクセスを制御する携帯電話向けの X Server を提案する．以後，この X Server をアクセス制御機能付き X Server と呼ぶ．X Client は X Server に X リクエストを送信し，X Server が受信した X リクエストを実行することで，X Server が管理する X リソースにアクセスする．そのため，X Server は X リクエストの実行，不実行を制御することで，X Client による X リソースへのアクセスを制御できる．この X リクエストの実行，不実行を制御することを X リクエストの実行制御と呼ぶ．

アクセス制御機能付き X Server は，NSA [4] が発表しているセキュア X Server[3)] のアクセス制御手法をベースとしている．セキュア X Server は SELinux 上での動作を前提としており，SELinux のプロセスに付与されるセキュリティ・コンテキストを用いてアクセス制御を行う．本研究で新たにアクセス制御機能を持つ X Server を提案する理由を以下に述べる．

- SELinux 非依存性
  アクセス制御機能付き X Server を動作させる OS は，SELinux が提供するようなアクセス制御機能を備えることが望ましい．セキュア X Server は，SELinux 上での動作を前提としているが，SELinux 以外のアクセス制御機能を備えた Linux 上で動作させるために，SELinux に非依存であることが望まれる．

- アプリケーション実行環境が要求する機能の追加
  携帯電話のアプリケーション実行環境は，PC や PDA のアプリケーション実行環境とは異なる要求を求められる．アクセス制御機能付き X Server は，この要求を満たすための機能を実現する必要がある．

[2] Linux は Linus Torvalds の商標または登録商標である．
[3] X Window System は X Consortium, Inc. の登録商標である．
[4] National Security Agency, 米国家安全保障局．

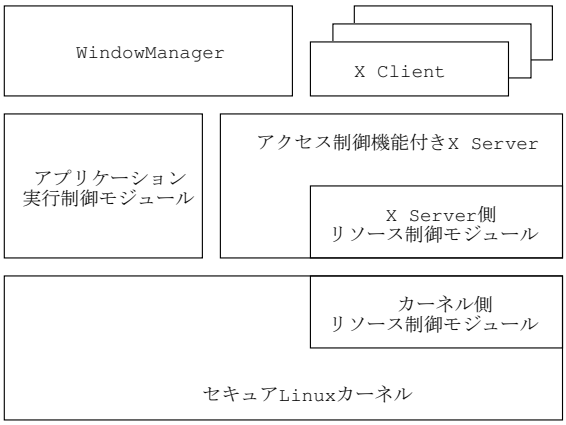

Fig. 1　アプリケーション実行環境

以後，2 章ではアクセス制御機能付き X Server を動作させる携帯電話のアプリケーション実行環境について述べる．3 章ではアクセス制御機能付き X Server に求められる要求機能について述べ，4 章では 3 章を踏まえて，アクセス制御機能付き X Server のアクセス制御手法について述べる．最後に，5 章でまとめと今後の予定を述べる．

2　携帯電話のアプリケーション実行環境

アクセス制御機能付き X Server を動作させる携帯電話のアプリケーション実行環境を Fig. 1 に示す．Fig. 1 に示したアプリケーション実行環境の構成要素を順に説明する．

- セキュア Linux カーネル
  悪意あるアプリケーションによりアクセス制御機能付き X Server や設定ファイルが改竄されると，アクセス制御機能付き X Server は改竄された設定にもとづくアクセス制御を行うため，正しいアクセス制御を行えない．通常の Linux では，セキュリティホールなどにより悪意あるアプリケーションが root 権限を取得すると，アクセス制御機能付き X Server のプログラムファイルや設定ファイルの改竄が可能になる．このような改竄を防ぐために，必要な権限のみをアプリケーションに付与するセキュア Linux カーネルが必要となる．また，電話帳などのユーザのプライバシーデー

タを保存しているという観点からも，セキュア Linux カーネルは必要である．

- リソース制御モジュール
  携帯電話のアプリケーション実行環境において，悪意あるアプリケーションからの攻撃を防ぐためには，アクセス制御だけでは不十分である．なぜなら携帯電話のリソース量は少ないため，リソースを消費する正規アクセスの繰り返しにより携帯電話のリソースを枯渇させることで，携帯電話が提供するサービスを妨害できるためである．この種の攻撃を防ぐために，カーネルと X Server はリソース制御モジュールを必要とする [4]．リソース制御モジュールは，アプリケーションごとにリソースの使用量を制限する．

- アプリケーション実行制御モジュール
  携帯電話のアプリケーション実行環境は，ユーザからの入力や着信などの外部イベントを起因として，画面に表示するアプリケーションの切り替えや起動を行う．携帯電話ではアプリケーションの切り替えや起動は厳密に管理され，PC や PDA のように任意の状態で任意のアプリケーションへの切り替えや起動はできない．アプリケーション実行制御モジュールは，アプリケーションの切り替えや起動などを管理する．

- アクセス制御機能付き X Server
  アクセス制御機能付き X Server は，X Client による X リソースへのアクセスを制御する．アクセス制御機能付き X Server と X Client との接続には，Unix ドメイン・ソケットを用いる．Unix ドメイン・ソケットを用いる利点は 2 つある．1 つ目の利点は，TCP/IP による接続と比較して性能が良い点である．2 つ目の利点は，X Server に接続した X Client をプロセス ID を用いて識別できる点である [5]．

- WindowManager
  WindowManager は，ウィンドウの装飾やウィンドウの表示を制御する．そのため WindowManager は，X Server と密接に連携して動作する．また，アプリケーション実行環境が要求する振る舞いを実現するために，ウィンドウの振る舞いを制御する．WindowManager がウィンドウの振る舞いを自由に制御するために，アクセス制御機能付き X Server が動作する環境の WindowManager は，他のアプリケーションのウィンドウに対する強いアクセス権限を必要とする．

3 要求機能

アプリケーション実行環境が携帯電話として振る舞うために，アプリケーション実行環境に求められる 2 つの要求がある [6]．これらの要求を満たすために，アプリケーション実行環境は次の 2 つの機能をアクセス制御機能付き X Server と WindowManager に要求する．

機能 1 指定したアプリケーションを画面に表示すること

機能 2 指定したアプリケーションがフォーカスを取得すること

上記 2 つの機能を実現するために，アプリケーションのウィンドウを画面に表示できる権限を示す表示権限と，アプリケーションがフォーカスを取得できる権限を示す入力権限を導入する．機能 1 を実現するためには，指定したアプリケーションに表示権限を与え，表示権限を持たないアプリケーションのウィンドウの表示を禁止する．機能 2 を実現するためには，指定したアプリケーションに入力権限を与え，入力権限を持たないアプリケーションによるフォーカスの取得を禁止する．これらの表示権限や入力権限を持つアプリケーションは，アプリケーション実行制御モジュールが指定する．

ウィンドウの表示は，他のアプリケーションのウィンドウとの相対的な位置やウィンドウのスタック順序により決まる．そのため，アクセス制御機能付き X Server がウィンドウにアクセスする個々の X リクエストを実行制御することでウィンドウの表示を制御するよりも，WindowManager が表示権限にもとづいてウィンドウの表示を制御する方が望ましい．

フォーカスの取得は，アプリケーションがフォーカス設定を要求する X リクエストを X Server に送信することで行う．そのため，アクセス制御機能付き X Server が入力権限にもとづいた X リク

エストの実行を制御することで，アプリケーションによるフォーカスの取得を制御する．

4　アクセス制御手法

アクセス制御機能付き X Server は，X リクエストの実行制御により X Client のアクセス制御を行う．アクセス制御機能付き X Server は X リクエストの実行制御を行うときに，X リクエストがアクセスするすべての X リソースとそれらの X リソースに対するアクセスの種類ごとに，X リクエストを分割する．以後，X リソースに対するアクセスの種類をオペレーションと呼び，”リソース名:アクセスの種類”という形式で表記する．オペレーションはセキュア X Server の permission に相当する．X リクエストを送信した X Client が X リクエストの実行に必要なすべてのオペレーションのアクセス権限を持つとき，アクセス制御機能付き X Server はその X リクエストを実行する．

4.1　構成

Fig. 2 にアクセス制御機能付き X Server の処理構成図を示す．アクセス制御機能付き X Server は X リクエストを受信すると，受信した X リクエストをオペレーションに分割する．そして，X リクエストを送信した X Client がオペレーションのアクセス権限を持つか調べる．X リクエストを送信した X Client がすべてのオペレーションのアクセス権限を持つとき，アクセス制御機能付き X Server は受信した X リクエストを実行する．

アクセス制御機能付き X Server は，X Client のアクセス権限をアクセス制御ルールを用いて調べる．アクセス制御ルールとは，X Client がアクセス可能な X リソースとそのオペレーションとの組みの集合を記述したものである．また，アクセス制御ルールを格納したデータベースをアクセス制御ルールデータベースと呼ぶ．

4.2　オペレーション

アクセス制御機能付き X Server がアクセス制御の対象とする X リソースを Table 1 に示す．X リソースは，プロセスリソースと共有リソースの 2 つに分類される．プロセスリソースは，そのリソースを作成した X Client により所有される．一方，共有リソースは，X Server により所有される．X リソースを所有する X Client，もしくは，X Server を X リソースの所有アプリと呼ぶ．アクセス制御

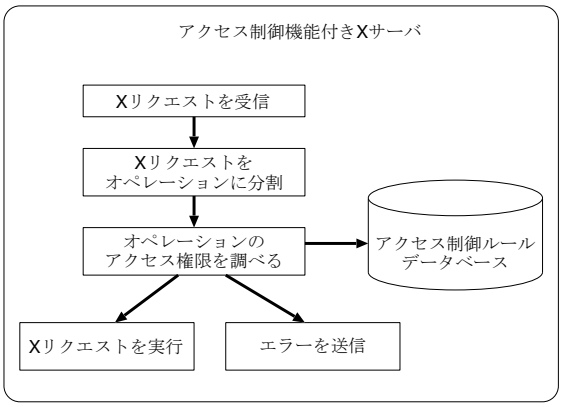

Fig. 2　処理構成図

Table 1　X リソース一覧

リソース名	種類
Client Window Drawable Colormap Cursor	プロセスリソース
Input Server Screen	共有リソース

機能付き X Server は，Table 1 に示したリソースごとにオペレーションを定義する．アクセス制御機能付き X Server が定義するオペレーションを Table 2 と Table 3 に示す．

4.3　X リクエスト

X Client が送信する X リクエストの種類は，119 種類である．X リクエストはその実行中に，0 個以上の X リソースにアクセスする．そのためアクセス制御機能付き X Server は，X リクエストがアクセスする X リソースとそのオペレーションの組みごとに X リクエストを分割する．すなわちアクセス制御機能付き X Server は，X リクエストを 0 個以上のオペレーションから構成されているとみなす．Table 4 に X リクエストとその X リクエストを構成するオペレーションと X リソースとの組みの一部を示す．

4.4　アクセス制御ルール

アクセス制御機能付き X Server は，起動時にアクセス制御ルールを記述したファイルを読み込み，アクセス制御ルールデータベースを構築する．アクセス制御ルールは，アクセス権を持つ X Client，オペレーション，オペレーションがアクセスする X

Table 2　プロセスリソースのオペレーション一覧

オペレーション	説明
Client:kill	X Client を終了する
Client:setclosedownmode	X Client 終了時のリソースの扱いを変更する
Window:addchild	子 Window を追加する
Window:destroy	Window を削除する
Window:map	Window を表示する
Window:unmap	Window を非表示にする
Window:chstack	子 Window のスタック順序を変更する
Window:chprop	Window のプロパティを変更する
Window:listprop	Window のプロパティを取得する
Window:getattr	Window の属性を取得する
Window:setattr	Window の属性を設定する
Window:move	Window の座標を変更する
Window:chselection	Selection を変更する
Window:chparent	Window の親を変更する
Window:ctrllife	Window の生存期間を制御する
Window:enumerate	子 Window を列挙する
Window:grab	Window の grab を制御する
Window:remove	子 Window を取り除く
Window:sendclientevent	クライアントイベントを送信する
Window:sendserverevent	サーバイベントを送信する
drawable:destroy	Drawable を削除する
drawable:draw	Drawable に描画する
drawable:copy	Drawable からピクセルをコピーする
drawable:getattr	Drawable の属性を取得する
Colormap:destroy	Colormap を削除する
Colormap:install	Colormap をインストールする
Colormap:uninstall	Colormap をアンインストールする
Colormap:alloccolor	Colormap に color cell を作成する
Colormap:store	Colormap の color cell に書き込む
Colormap:freecolor	Colormap から color cell を削除する
Cursor:destroy	カーソルを削除する
Cursor:assign	カーソルと Window を関連付ける
Cursor:chattr	カーソルの属性を変更する

Table 3　共有リソースのオペレーション一覧

オペレーション	説明
Input:getattr	入力デバイスの属性値を取得する
Input:setattr	入力デバイスに属性値を設定する
Input:grab	グラブを取得する
Input:passivegrab	パッシブグラブを取得する
Input:ungrab	グラブを解放する
Input:passiveungrab	パッシブグラブを解放する
Input:bell	ベルを鳴らす
Input:mousemotion	マウスモーションイベントを取得する
Input:warppointer	マウスポインタを指定位置に移動させる
Input:focus	フォーカスを取得する
Server:screensaver	スクリーンセーバを制御する
Server:hostcontrol	ホストのアクセスコントロールを制御する
Server:setfontpath	フォントパスを設定する
Server:grab	X Server をロック，アンロックする
Server:createatom	Atom を作成する
Screen:installcolormap	Colormap をインストールする
Screen:uninstallcolormap	Colormap をアンインストールする
Screen:listcolormap	Colormap のリストを取得する
Screen:nobackground	透明な背景を持つ Window を作成する

Table 4　X リクエストを構成するオペレーション (抜粋)

X リクエスト	オペレーション	X リソース
CreateWindow	Window:addchild	親 Window
	Drawable:copy	使用 Pixmap
	Cursor:assign	使用 Cursor
ConfigureWindow	Window:setattr	対象 Window
	Window:move	対象 Window
	Window:chstack	親 Window
CreatePixmap	なし	なし
SetInputFocus	Input:focus	X Server
InternAtom	Server:createatom	X Server

リソースの所有アプリの 3 組で構成される．以後，アクセス権を持つ X Client をソースアプリ，オペレーションがアクセスする X リソースの所有アプリをターゲットアプリと呼ぶ．

次に，アクセス制御ルールの記述方法について述べる．アクセス制御ルールをソースアプリ単位，ターゲットアプリ単位で記述すると，アクセス制御ルールは X Client の数に比例して大きくなり，ルール記述の複雑化を招いてしまう．そのため，同じアクセス制御ルールを持つアプリケーションの集合をドメインとして定義し，ソースアプリとターゲットアプリの代わりにドメインを用いて記述する．また，ドメインに属するアプリケーションの情報も記述する．ドメインを用いて記述することで，X Client が持つアクセス制御ルールの違いに着目した記述が可能になり，アクセス制御ルールの記述を簡易化できる．

Fig. 3 にアクセス制御ルールの記述例を示す．Fig. 3 は，WM ドメインに WindowManager が，XServer ドメインに X Server が，Manufacturer ドメインに WEB ブラウザやメーラなどのビルトインアプリケーションが属する例である．アクセス制御ルールの記述を簡易にするために，任意のオペレーションや任意のアクセスの種類を表す”∗”の記述を用いている．

4.5　X Server によるアクセス制御

アクセス制御機能付き X Server は X リクエストを受信すると，以下の手順に従って X リクエストの実行制御を行う (Fig. 4)．

1. オペレーションへの分割
   受信した X リクエストを，その X リクエストを構成するオペレーションに分割する．このとき，X リクエストがアクセスする X リソースに対するオペレーションのみを抽出する．例えば Table 4 に挙げた CreateWindow リクエストにおいて Cursor を割り当てないときは，Cursor:assgin オペレーションは必要としない．

```
domain WM {
    {XServer, *}
    {Manufacturer, *}
}

domain XServer {
    {Manufacturer, *}
}

domain Manufacturer {
    {XServer, Server:createatom}
    {XServer, Window:addchild}
    {XServer, Window:remove}
}

[WM] /usr/X11R6/bin/wm
[XServer] /usr/X11R6/bin/xserver
[Manufacturer] /usr/bin/browser, /usr/bin/mailer
```

Fig. 3　アクセス制御ルールの記述例

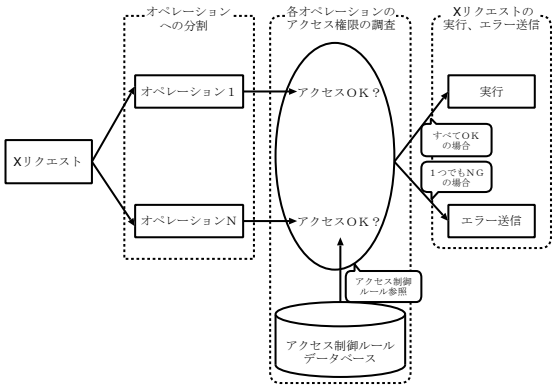

Fig. 4　作業流れ図

2. オペレーションのアクセス権限の調査
   1 で分割したオペレーションごとに，オペレーションのアクセス権限の有無を判定する．以下のいずれかの条件を満たしたとき，オペレーションのアクセス権限はあると判定する．

   条件 1 ソースアプリ，ターゲットアプリ，オペレーションの 3 組がアクセス制御ルールデータベースに登録されている

   条件 2 ソースアプリとターゲットアプリが一致する

   条件 2 は，自アプリの作成した X リソースには自由にアクセスできることを意味する．

3. X リクエストの実行とエラー送信
   2 において，分割したすべてのオペレーションのアクセス権限があると判定したとき，アクセス制御機能付き X Server は，X リクエストを実行する．また，1 において X リクエストを 0 個のオペレーションに分割した場合も，X リクエストを実行する．Table 4 の CreatePixmap リクエストがこの事例にあたる．逆にアクセス権限のないオペレーションが存在するとき，X リクエストを送信した X Client にエラーイベントを送信する．

4.6　X Server によるフォーカス取得制御

3 章で述べたように，フォーカスの取得はアクセス制御機能付き X Server により制御される．アプリケーションがフォーカスを取得するとき，SetInputFocus リクエストを X Server に送信する．そのため，SetInputFocus リクエストの実行を制御することで，アプリケーションによるフォーカスの取得を制御できる．Table 4 より，SetInputFocus リクエストの実行には X Server に対する Input:focus オペレーションのアクセス権限が必要となる．したがって，Input:focus オペレーションのアクセス権限を変更することで，SetInputFocus リクエストの実行を制御できる．

アクセス制御機能付き X Server は，アプリケーション実行制御モジュールにより入力権限を持つアプリケーションを指定されると，そのアプリケーションに Input:focus オペレーションのアクセス権限を与える．

4.7　WindowManager によるウィンドウ表示制御

3 章で述べたように，ウィンドウの表示は WindowManager により制御される．WindowManager は，ルートウィンドウの子ウィンドウにあたるトップレベルウィンドウのスタック順序を制御することにより，ウィンドウの表示を制御する [6]．アプリケーションがウィンドウのスタック順序を変更すると，X Server は WindowManager に CirculateRequest イベント，または，ConfigureRequest イベントを送信する．WindowManager はこれらのイベントの受信時に，ウィンドウのスタック順序を制御できる．

WindowManager は，表示権限を持たないアプリケーションのトップレベルウィンドウをスタック順序の上位に配置するスタック順序変更を抑制する．このようなスタック順序変更の抑制により，表示権限を持つアプリケーションのトップレベル

ウィンドウをスタック順序の上位に常に配置するように制御する．

4.8 セキュア X Server との相違点

アクセス制御機能付き X Server はセキュア X Server のアクセス制御手法をベースとしているが，携帯電話向けに修正を加えている．アクセス制御機能付き X Server とセキュア X Server との主要な相違点を以下に示す．

- create 系オペレーションの削除
  アクセス制御機能付き X Server は，Window や Drawable などの X リソースの create 系のオペレーションを定義していない．つまりアプリケーションは，一部の例外を除き自由に X リソースを作成できる．削除した理由は，X リソースの作成がセキュリティの侵害にならないからである．ただし，リソースを枯渇させる攻撃を防ぐために，2 章に述べたリソース制御モジュールが必要となる．

- Font リソース系オペレーションの削除
  アクセス制御機能付き X Server は，Font リソース系のオペレーションを定義していない．削除した理由は，携帯電話にインストールされているフォントの数は少なく，フォントの利用を制限する必要が無いからである．

- SELinux からの独立
  アクセス制御機能付き X Server は，通常の Linux 上で動作する．しかし，アクセス制御機能付き X Server や WindowManager を安全に動作させるために，Linux にはなんらかのアクセス制御機能が必要である．

- Atom 作成オペレーションの追加
  Atom は，全 X Client が共有する X リソースである．そのため，リソース制御モジュールは，Atom のメモリを X Server の消費メモリに割り当てるのが妥当である．その結果，X Client が無制限に Atom を作成すると，X Server のメモリが枯渇してしまう．X Server のメモリ枯渇を防ぐために，Atom 作成のオペレーション (Server:createatom) を追加し，Atom の作成を制御する必要がある．

- 入力権限，表示権限の導入
  セキュア X Server は，WindowManager が入力デバイスのラベルを変更することで，フォーカスの取得を許可する X Client を制御している．またセキュア X Server には，表示権限に相当する概念は存在しない．

  一方，アクセス制御機能付き X Server は，フォーカスの取得を制御するためにアクセス権限を導入した．また，WindowManager がウィンドウ表示を制御するために，表示権限を導入した．

- アクセス制御ルール記述の違い
  セキュア X Server は SELinux 上での動作を想定しているため，アクセス制御ルールを SELinux のアクセス制御ルールを記述するポリシーファイルに記述する．

  一方，アクセス制御機能付き X Server は，SELinux 以外の Linux 上でも動作するように，アクセス制御ルールを独自のファイルに記述する．

- エラーの送信
  セキュア X Server は，アクセス制御により X リクエストを実行しない場合，X Client にエラーを送信しない．

  一方，アクセス制御機能付き X Server は，X リクエストを実行しない場合，X Client にエラーを送信する．X Client はエラーの受信により，X リクエストの実行の失敗を知ることができる．X Client はエラーを適切に処理することで意図しない動作を防ぐことができ，ユーザビリティの向上を図れる．

5 おわりに

本稿では，携帯電話向けのアクセス制御機能付き X Server を提案した．アクセス制御機能付き X Server は，セキュア X Server のアクセス制御手法をベースとしており，X Client から受信した X リクエストの実行をアクセス制御ルールにもとづき制御する．さらに，アプリケーション実行環境が携帯電話として振る舞うために必要な要求を満たすために入力権限と表示権限を導入した．アクセス制御機能付き X Server と WindowManager は，これらの権限にもとづきアプリケーションの画面への表示やフォーカス取得を制御する．また，ア

クセス制御機能付き X Server とセキュア X Server との相違点について述べた．

アクセス制御機能付き X Server は，X Client による X リソースへのアクセスを制御するのみである．携帯電話のアプリケーション実行環境のセキュリティを強化するためには，X Server のアクセス制御だけでは不十分であり，他のサーバアプリケーションに対しても同様なアクセス制御が求められる．

現在，アクセス制御機能付き X Server は未実装である．今後，アクセス制御機能付き X Server を実装し，以下に挙げる評価を行う必要がある．

- 性能
  アクセス制御機能付き X Server はすべての X リクエストの実行制御を行うため，X リクエストの実行速度が低下してしまう．この実行速度の増分を定量的に求め，実用性の評価が必要である．

- アクセス制御ルールの記述の容易性
  アクセス制御機能付き X Server は，アクセス制御ルールにもとづき X リクエストの実行制御を行う．そのため，アクセス制御ルールの正確な記述が求められる．アクセス制御ルールの記述が複雑であると，記述ミスの増加や，記述ミスの発見が困難になり，正確な記述が難しくなる．よって，アクセス制御ルールの記述の難易度の評価が必要である．

- オペレーションの粒度
  X リクエストの実行制御は，X リクエストを構成するオペレーションのアクセス制御により実現される．今回，X リクエストを構成するオペレーションを定義したが，実運用時により細かいアクセス制御を実現するために，新たなオペレーションの追加が必要となる可能性もある．また，悪意ある X Client による想定外の攻撃を防ぐために，X リクエストを構成するオペレーションの種類の再構成が必要となる可能性もある．そのため，実際の携帯電話のアプリケーション実行環境において，オペレーションの粒度の妥当性の評価が必要である．

参考文献

1) BREW, http://www.qualcomm.com/brew

2) Symbian, http://www.symbian.com/

3) Doug Kilpatrick, Wayne Salamon and Chris Vance, ”Securing The X Window System With SELinux”, http://www.nsa.gov/selinux/papers/X11_Study.pdf

4) 稗田論士, 才田好則, 中山義孝, 千嶋博, 中本幸一, ”携帯端末向け Linux のリソース管理の実現”, 情処学論, vol.46, no.SIG3, pp.1-11, Jan. 2005.

5) David A. Wheeler, ”Secure Programming for Linux and Unix HOWTO”, http://www.linux.org/docs/ldp/howto/Secure-Programs-HOWTO/index.html

6) Yoshiharu Asakura, Gen Okuyama, Yoshitaka Nakayama, Kazutoshi Usui and Yukikazu Nakamoto, ”Multi-Application Platform for Mobile Phones”, Proc. of WSTFEUS 2004, IEEE, pp.139-143, May 2004.

7) Oliver Jones, ”Introduction to The X Window System”, PRENTICE HALL, 1989.