(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5183978号
(P5183978)

(45)発行日　平成25年4月17日(2013.4.17)　(24)登録日　平成25年1月25日(2013.1.25)

(51)Int.Cl.　　F I
G06F 21/44 (2013.01)　　G06F 21/20 144C
H04L 9/32 (2006.01)　　H04L 9/00 675A

請求項の数 8　(全 9 頁)

(21)出願番号　特願2007-157637 (P2007-157637)
(22)出願日　平成19年6月14日 (2007.6.14)
(65)公開番号　特開2007-336558 (P2007-336558A)
(43)公開日　平成19年12月27日 (2007.12.27)
審査請求日　平成22年6月2日 (2010.6.2)
(31)優先権主張番号　06290992.4
(32)優先日　平成18年6月16日 (2006.6.16)
(33)優先権主張国　欧州特許庁 (EP)

(73)特許権者　501263810
トムソン　ライセンシング
Ｔｈｏｍｓｏｎ　Ｌｉｃｅｎｓｉｎｇ
フランス国，　９２１３０　イツシー　レ　ムーリノー，　ル　ジヤンヌ　ダルク，　１－５
１－５，　ｒｕｅ　Ｊｅａｎｎｅ　ｄ’Ａｒｃ，　９２１３０　ＩＳＳＹ　ＬＥＳ　ＭＯＵＬＩＮＥＡＵＸ，　Ｆｒａｎｃｅ
(74)代理人　100070150
弁理士　伊東　忠彦
(74)代理人　100091214
弁理士　大貫　進介
(74)代理人　100107766
弁理士　伊東　忠重

最終頁に続く

(54)【発明の名称】エミュレートされたクライアントを発見する高精度非サイクル測定を用いた装置及び方法

(57)【特許請求の範囲】
【請求項１】
クライアントがエミュレートされているかを発見する方法であって、
チャレンジを前記クライアントに送出する工程であって、解くために決定論的な数の反復を前記チャレンジが必要とする工程と、
前記チャレンジの送出と応答の受信との間の時間を測定する工程と、
前記チャレンジを解くために必要な数の反復を備えた応答を受信すると、前記応答における前記反復の数が反復の期待数に一致した場合であり、かつ、前記応答がタイムリーであった場合に、前記クライアントはエミュレートされていないと判定する工程とを備える方法。
【請求項２】
請求項１記載の方法であって、前記チャレンジは、前記クライアントの識別情報又は前記クライアントのユーザの識別情報に依存している方法。
【請求項３】
請求項１記載の方法であって、前記チャレンジが暗号チャレンジである方法。
【請求項４】
請求項３記載の方法であって、前記暗号チャレンジが鍵サーチである方法。
【請求項５】
請求項４記載の方法であって、前記暗号チャレンジが、値の非暗号化バージョンと、前記値の暗号化バージョンと、前記鍵サーチにおいて第１の鍵が用いるための基底とを備え

る方法。
【請求項６】
クライアントがエミュレートされているかを発見する検証器であって、
チャレンジを前記クライアントに送出し、前記クライアントから応答を受信する通信インタフェースであって、解くために決定論的な数の反復を前記チャレンジが必要とし、前記応答は、前記チャレンジを解くために必要な反復の数を備えている通信インタフェースと、
前記チャレンジの送出と前記応答の受信との間の時間を測定するタイマと、
前記チャレンジを選択し、
前記応答内の反復の数が反復の期待数に一致する場合であり、かつ、前記応答もタイムリーである場合に、前記クライアントがエミュレートされていないことを判定する、
プロセッサと
を備える検証器。
【請求項７】
請求項6記載の検証器であって、前記プロセッサを、前記クライアントの識別情報又は前記クライアントのユーザの識別情報に前記チャレンジを依存させるよう適合させた検証器。
【請求項８】
請求項6記載の検証器であって、前記プロセッサを、値の非暗号化バージョン、前記値の暗号化バージョン、及び前記鍵サーチにおいて第１の鍵が用いるための基底を備える暗号鍵サーチ・チャレンジであるチャレンジを選択するよう適合させた検証器。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般にセキュア・ネットワークに関し、特に前述のネットワークにおける装置の認証に関する。
【背景技術】
【０００２】
クライアントにサービスを提供するサーバを有することがネットワークにおいて非常に一般的である。通常はユーザにあるクライアントはサービスをサーバから要求する。このサーバは、通常、サービスにアクセスする権利をクライアントが有している旨の検証後にサービスをクライアントに提供する。前述のサービスの例には、ビデオオンデマンド、文書の印刷、及び施錠されたドアの開錠がある。
【０００３】
残念ながら、前述のサービスへのアクセスを得ることを、そうする権利を有することなしに試行する人々（いわゆる「ハッカー」や「海賊」）が存在している。この目的で、前述の人々は、ネットワーク内のセキュリティ解決策をだしぬくために種々の手法を用いる。
【０００４】
ハッカーの活動をくじこうとする解決策の１つに認証がある。すなわち、サーバは、真のクライアントであることを確実にするためにクライアントを認証する。認証は、有効な、又は署名された識別情報の提示、ユーザ名及び関連したパスワードの提示、又は対称暗号法又は非対称暗号法が関係するプロトコルを用いて行うことができる。
【０００５】
別の解決策は、特定のネットワーク内に受信装置があることを確実することである。例えば、欧州特許第１６５０６７１号明細書には、適切な応答を算出する受信器に、ランダムなチャレンジを送信器が送出するシステムが開示されている。次いで、送信器は、応答を戻すよう受信器に指示し、応答の受信までの時間を算出する。しかし、前述の解決策は、受信器までの距離を送信器が算出することを可能にするだけであるが、受信器の特性（送信器が事情により通じていなくても十分エミュレートすることができる）に関しては全

く確実性を与えない。
【０００６】
更に別の解決策は、その秘密を回復するか又はその挙動を修正するためにリバースエンジニアリングすることが難しいそのクローズド・プラットフォームを用いることである。認証などの他の解決策とともに通常用いるこの解決策は例えば、ゲーム・ステーション（例えば、プレイステーションやXボックス）、有料TVシステムのセットトップ・ボックス、トリプルプレイ・モデム（例えば、フリーボックスやライブボックス）や携帯電話機に用いられている。これは当然、アーキテクチャの多様性にその強みが起因するということができるパソコン（PC）とはかなり異なる。
【０００７】
サーバを模倣するのに十分であるようにクローズド・プラットフォームをエミュレートすることが難しい一方、これが不可能でないことが明らかになった。この課題に対する標準的な解決策は取り消しである。クライアントが破られたことをシステム権限が知った場合、そのクライアントは、取り消しリストに載せられる。認証中、サーバはまず、クライアントが取り消しリストに載っているかを検証し、肯定の場合、サービスを拒否する。
【０００８】
取り消しが効率的であるには、システム権限は、装置が破られたことを知っている必要がある。このことには長い時間がかかり得る。この間、ハッカーは、サービスを享受し続けるか、又は、更に悪いことには、多くの人々がサービスを利用することができるように装置をエミュレートする方法を他の人々に知らせることができる。
【０００９】
この課題に対する解決策は、例えば、実行時間を測定することによってクライアントのフィンガープリンティングを行うことである。前述の種類のフィンガープリンティングは、エミュレートされたクライアントが、本物のプラットフォーム上のクライアントよりも遅いことを前提とする。しかし、プロセッサがなお一層高速化するにつれ、このことは、依存する対象の前提ではない（特に特定の年齢のプラットフォームの場合）。
【発明の開示】
【発明が解決しようとする課題】
【００１０】
よって、現行のセキュリティ解決策を改良し、エミュレートされた装置の検出をそれによってより容易にする解決策に対する必要性が存在している。
【課題を解決するための手段】
【００１１】
本発明は前述の解決策を提供する。
【００１２】
第１の局面では、本発明は、クライアントがエミュレートされたかを発見する方法に関する。解くために決定論的な数の反復を必要とするチャレンジがクライアントに送出される。上記数の反復を備えた応答を受信すると、応答における反復の数が反復の期待数に一致した場合、クライアントはエミュレートされていないと判定される。
【００１３】
好ましい実施例では、チャレンジの送出と応答の受信との間の時間が測定され、応答がタイムリーであることをやはり前提とすれば、クライアントがエミュレートされていないことが判定される。
【００１４】
好ましい実施例では、チャレンジは、クライアントの識別情報又はクライアントのユーザの識別情報に依存している。
【００１５】
別の好ましい実施例では、チャレンジは暗号チャレンジであり、効果的には鍵サーチである。暗号チャレンジが、値の非暗号化バージョンと、値の暗号化バージョンと、鍵サーチにおいて第１の鍵が用いるための基底とを備えることは特に有利である。

【００１６】
第２の局面では、本発明は、クライアントがエミュレートされたかを発見する検証器に関する。検証器は、クライアントにチャレンジを送出する通信インタフェースを備える。チャレンジは、解くために決定論的な数の反復を必要とする。インタフェースは、上記反復数を備えた応答をクライアントからの受信するためでもある。検証器は、チャレンジを選択し、応答内の反復数が、反復の期待数に一致する場合に、クライアントがエミュレートされていないことを判定するプロセッサを更に備える。
【００１７】
好ましい実施例では、検証器は、チャレンジの送出と応答の受信との間の時間を測定するタイマも備え、プロセッサは、クライアントがエミュレートされていないことを判定するために、応答がタイムリーであったことを検証するようにも適合させる。
【００１８】
好ましい実施例では、プロセッサは、クライアントの識別情報又はクライアントのユーザの識別情報にチャレンジを依存させるよう適合させる。
【００１９】
更に別の好ましい実施例では、値の非暗号化バージョン、その値の暗号化バージョン、及び鍵サーチにおいて第１の鍵が用いるための基底を備える暗号鍵サーチ・チャレンジであるチャレンジを選択するよう適合させる。
【発明を実施するための最良の形態】
【００２０】
本発明の好ましい実施例を、例として、添付図面を参照して次に説明する。
【実施例】
【００２１】
図１は、互いに相互作用するよう適合させたクライアント１０及び検証器２０を示す。クライアント１０は、プロセッサ（CPU）１１と、プロセッサ１１によって実行されるアプリケーションを記憶するよう適合させたメモリ１３と、装置、特に検証器２０とネットワーク１を介して通信するよう適合させた通信インタフェース（I／O）１４とを備える。検証器２０は、プロセッサ（CPU）２１と、プロセッサ２１専用のタイマ２２と、プロセッサ２１によって用いるメモリ２３と、装置、特にクライアント１０とネットワーク１を介して通信するよう適合させた通信インタフェース（I／O）２４とを備える。
【００２２】
図２は、本発明の好ましい実施例によって、クライアントが、特定のプラットフォーム上で実行する（すなわち、エミュレートされていない）ことを検証する方法の好ましい実施例を示す。クライアント１０が特定のプラットフォーム上で実行することを検証器２０が検証したい場合、クライアント１０との通信セッションを開設する（２０２）。検証器は次いで、クライアント１０の暗号チャレンジCを好ましくはランダムに選択する（２０４）。暗号チャレンジCは、暗号関数を用いて（好ましくは、鍵長が128ビットの高度暗号化標準（AES）アルゴリズムに基づいて）計算される。好ましい実施例では、暗号チャレンジCは、（clear,start,$\{clear\}_{start+tries}$）として表すことができる。ここで、
clearは乱数、
startは、暗号チャレンジCの始点として用いられる乱数、
triesは、クライアント１０によって行う対象の試行の数である。すなわち、start+triesが、暗号チャレンジCを破るための終点として用いられる。
【００２３】
$\{clear\}_{start+tries}$は、鍵数start+triesを備えたclearの暗号化の結果である。これは、暗号チャレンジCの暗号化メッセージである。
【００２４】
検証器２０は、選ばれた暗号チャレンジCをクライアント１０に送出し（２０６）、タイマ２２を開始する（２０８）。暗号チャレンジCを受信すると、クライアント１０はこれをそのメモリ１３に記憶し、所定のアルゴリズムに従って、正しい鍵が求められるまで

鍵を順次試行して、決定論的鍵サーチを行う（２１０）。好ましい鍵サーチ・アルゴリズムでは、

1. 試行カウンタi=1を初期化する
2. $\{clear\}_{start+i}$を計算する
3. 結果が$\{clear\}_{start+tries}$に等しいか否かを確認する

4a. 肯定の場合、暗号チャレンジCが破られた結果としてiを検証器２０に送出する（２１２）

4b. 否定の場合、iを増やし、工程２に進む。

【００２５】

多くの変形（例えば、iを修正するための単純な増加よりも複雑な関数を用いることなど）が可能である。別の実施例において、工程4aでは、クライアント１０は更に、検証器２０に対するメッセージにおいて、正しい鍵を含める。

【００２６】

別の好ましい実施例では、暗号チャレンジCは、$(clear, K_{start}, \{clear\}_{K=H(tries)})$として表すことができる。ここで、

clearは、クライアント１０又はユーザに応じた数又はデータ（乱数との排他的論理和（XOR）がとられたクライアント識別番号、又はユーザ識別番号若しくはユーザ加入番号など）である。

【００２７】

K_{start}は、鍵サーチの始点として用いるための鍵である。

【００２８】

$\{clear\}_{K=H(tries)}$は、試行の（すなわち、正しい鍵を求めるのに必要な反復の数の）ハッシュ値に対応する鍵によって暗号化されたclearである。乱数との排他的論理和（XOR）がとられた試行の数のハッシュ値に対応する鍵Kを用いることも可能である。

【００２９】

別の好ましい実施例の鍵サーチ・アルゴリズムでは、

1. 試行カウンタi=1を初期化する
2. $K_i=K_{start}+i$を計算する
3. $\{clear\}_{Ki}$を計算する
4. 結果が$\{clear\}_{H(tries)}$に等しいか否かを確認する

5a. 肯定の場合、暗号チャレンジCが破られた結果としてK_i及び試行数iを検証器２０に送出する

5b. 否定の場合、iを増やし、工程２に進む。

【００３０】

更に別の好ましい実施例では、暗号チャレンジCは、$(clear, K_{start}, \{clear\}_{K=Htries(Kstart)})$として表すことができる。ここで、

clearは、クライアント１０又はユーザに応じた数又はデータ（乱数との排他的論理和（XOR）がとられたクライアント識別番号、又はユーザ識別番号若しくはユーザ加入番号など）である。

【００３１】

K_{start}は、鍵サーチの始点として用いるための鍵である。

【００３２】

$\{clear\}_{K=Htries(Kstart)}$は、$K_{start}$をtries回ハッシングすることによって算出される値に対応する鍵によって暗号化されたclearである。

【００３３】

別の好ましい実施例の鍵サーチ・アルゴリズムは、

1. i=1及び$K_0=K_{start}$に初期化する
2. $K_i=H(K_{i-1})$を計算する
3. $\{clear\}_{Ki}$を計算する
4. 結果が$\{clear\}_{Htries(Kstart)}$に等しいか否かを確認する

5a. 肯定の場合、暗号チャレンジCが破られた結果としてK_i及び試行数iを検証器２０に送出する

5b 否定の場合、iを増やし、工程２に進む。

【００３４】

更に別の実施例では、暗号チャレンジは、クライアント１０がハッシュ関数を逆にすることを必要とする。

【００３５】

クライアント１０から結果２１２を受信すると、検証器２０はそのタイマ２２を停止させる（２１４）。すなわち、検証器のタイマは、チャレンジの送出と応答の受信との間の時間を測定する。検証器２０は次いで、結果（すなわち、start+i）が期待値（すなわち、start+tries）に等しいこと（すなわち、iがtriesに等しいこと）を確認する（２１６）。

【００３６】

確認２１６が首尾良く行われなかった場合、検証器２０は、クライアント１０がエミュレートされていることが分かる。しかし、確認２１６が首尾良く行われた場合、検証器２０は、そのタイマ２２から読み出された値が、選ばれたネットワーク送信遅延を期待時間に加えたものを超えないことの確認（２１８）に進む。すなわち、検証器２０は、応答２１２がタイムリーであることを確認する。それによって、適切な応答をハッカーが破るのに利用可能な時間が削減される。この第２の確認２１８が首尾良く行われた場合、クライアント１０が期待プラットフォーム上で実行することを検証器２０は結論付け得る。

【００３７】

別の実施例では、検証器２０は、期待プラットフォーム上で実行することを結論付ける前に、連続したいくつかのチャレンジにクライアント１０が正しく応答することを必要とし得る。

【００３８】

好ましい実施例では、クライアント１０又はユーザにリンクした識別子やその他のデータ（例えば、排他的論理和（XOR）、加算、ハッシング又は暗号化によって乱数と組み合わせられたクライアント識別番号、又はユーザ識別番号若しくはユーザ加入番号など）である。これによって、ハッカーが、例えば２つのクライアント（すなわち、盗んだIDによってエミュレートされた一クライアント、及びエミュレートされていない一クライアント）を用いて、それらの一方に対する第１の検証工程及び他方に対する第２の工程を通過させることがより難しくなる。

【００３９】

暗号チャレンジを識別子に依存させるために、サーバは好ましくは、start+triesの代わりにstart+tries+（識別子$\mathrm{mod}2^{32}$）を用いて前述のチャレンジを作成する。サーバは次いで、結果を算出するが、start+triesのみをクライアントに送出する。それは、その識別子$\mathrm{mod}2^{32}$によってこれを完了する。よって、チャレンジが、エミュレートされていないプラットフォームに対して、正しい識別子を用いて行われたことが検証される。

【００４０】

本発明は、暗号チャレンジの使用に限定されないが、クライアントが所定の手法によって算出を行い、必要な反復数を戻すことを必要とする他の種類のチャレンジを用いることもできる。例えば、特定の開始値から始まる、数学的関数の特定の精度までの反復的算出。

【００４１】

本発明によって、ハッカーにとって一層難しくなるように適合させることができる、エミュレートされたクライアントの費用効果の高い検出が可能になる。

【００４２】

本発明は例として説明しているに過ぎず、詳細の修正は、本発明の範囲から逸脱することなく行うことが可能である。

本明細書、並びに（該当する場合）特許請求の範囲、及び添付図面に記載された各構成

は、別個に備えても、何れの適切な組み合わせで備えてもよい。ハードウェアで実現されるものとして記載した構成は、ソフトウェアによっても実現することができ、逆も同様である。接続は適宜、無線接続又は有線接続（必ずしも直接又は専用でなくてよい）として実現することができる。

【００４３】

特許請求の範囲記載の参照符号は、例証の目的に過ぎず、特許請求の範囲記載の範囲を限定する効果を何ら有しないものとする。

【図面の簡単な説明】

【００４４】

【図１】本発明の好ましい実施例によるクライアント及び検証器を示す図である。

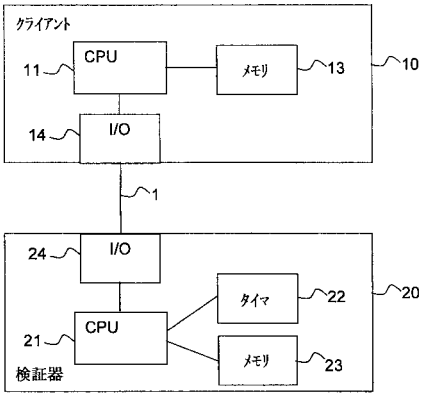
【図２】本発明の好ましい実施例による、特定のプラットフォーム上でクライアントが実行することを検証する方法を示す図である。

【符号の説明】

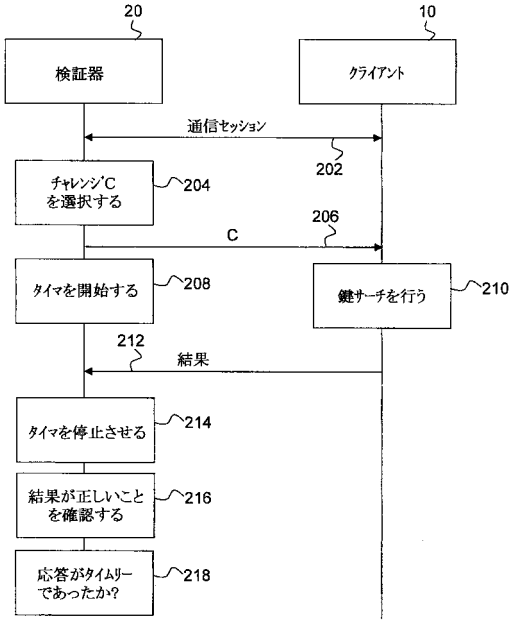
【００４５】

１　ネットワーク
１０　クライアント
１１　CPU
１３　メモリ
１４　I／O
２０　検証器
２１　CPU
２２　タイマ
２３　メモリ
２４　I／O
２０２　通信セッション
２０６　チャレンジ
２１２　結果

【図１】


クライアント
CPU
11
メモリ
13
10
14
I/O
1
24
I/O
タイマ
22
20
CPU
21
メモリ
23
検証器

【図２】


20
10
検証器
クライアント
通信セッション
202
チャレンジ'C
を選択する
204
206
C
タイマを開始する
208
鍵サーチを行う
210
212
結果
タイマを停止させる
214
結果が正しいこと
を確認する
216
応答がタイムリー
であったか?
218

フロントページの続き

(72)発明者 オリヴィエ イーン
フランス国, ３５４１０ ドンルー, リュ・デ・トゥルヌソル ２０
(72)発明者 エリック ディエル
フランス国, ３５３４０ リフレ, ラ・ビュザールディエール（番地なし）
(72)発明者 アラン デュラン
フランス国, ３５０００ レンヌ, リュ・ド・ディナン ７９
(72)発明者 モハメッド カルミ
フランス国, ３５２００ レンヌ, リュ・デュ・グヴェルヌール・ジェネラル・フェリックス・エブエ ８
(72)発明者 ニコラ プリジャン
フランス国, ３５０００ レンヌ, リュ・アンリ・ル・ギユー ３１

審査官 中里 裕正

(56)参考文献 特開２００６－１２７５２１（ＪＰ, Ａ）
米国特許第０７１９７６３９（ＵＳ, Ｂ１）
Juels, A. and Brainard, J., Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks, Proceedings of the 1999 Network and Distributed System Security Symposium, １９９９年, ＵＲＬ, http://www.isoc.org/isoc/conferences/ndss/99/
Kennell, R. and Jamieson, L. H., Establishing the Genuinity of Remote Computer Systems, Proceedings of the 12th USENIX Security Symposium, ２００３年, p.295-310, ＵＲＬ, http://static.usenix.org/events/sec03/tech/kennell/kennell.pdf

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｆ ２１／２０
Ｈ０４Ｌ ９／３２
ＪＳＴＰｌｕｓ／ＪＭＥＤＰｌｕｓ／ＪＳＴ７５８０（ＪＤｒｅａｍＩＩ）