ソフトウェア流通方式に関する一考察

1T－7

田中　利清
NTT 情報通信研究所

1．はじめに

ソフトウェアの配布・課金の方式としては、①無料配布、②売り切り、③使用量課金、の３つの方式が考えられる。①の例としては、使用期間や使用目的を限定したインターネットでのダウンロードなどがある。②の例としては、パッケージ販売や、暗号化したソフトウェアを CD-ROM 等で配布し復号鍵をインターネット等で販売するシステムなどがある。③の例としては、森らによりソフトウェア等の使用量に対して課金を行う超流通方式が提案され[1]、超流通方式を実現する研究が続けられている。

超流通方式の特徴の１つは、超流通専用ハードウェアをソフトウェア実行環境に設置し、ソフトウェア使用記録の作成・管理を行うことであるが、この専用ハードウェアの必要性が超流通方式の普及を阻害する要因の１つと考えられる。本論文では、専用ハードウェアを用いず、ソフトウェアのみにより、不正使用の攻撃に対してある程度の耐性を持つソフトウェア使用量課金を実現する方法を提案する。

2．ソフトウェア流通モデル

ソフトウェア流通方式を検討するに当たり、そのモデルを下記のように設定する。（図１）

① ソフトウェア実行マシン（ユーザ・マシン）と課金サーバーがインターネット等のネットワークで相互接続されている。

② ソフトウェア実行マシン上に流通クライアントが存在し、ソフトウェアの実行制御および課金サーバーとの通信を実行する。

③ ソフトウェアは、ネットワークまたは CD-ROM 等の媒体を経由して、流通サーバーからユーザへ配布され、ソフトウェア実行マシンに格納される。

④ ソフトウェアの実行は、ソフトウェア実行マシン上の流通クライアントから課金サーバーへ使用要求を送付し、課金サーバーにおいて課金処理を行った後、復号鍵などのソフトウェア実行に必要な情報を課金サーバーから流通クライアントへ送付することにより行われる。

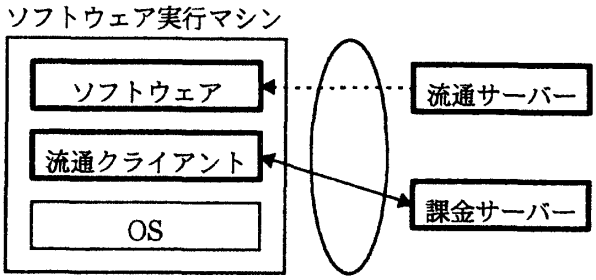

図１　ソフトウェア流通モデル

3．不正使用防止方法

インターネット等のネットワークまたは CD-ROM 等の媒体を経由した、流通サーバーからユーザへのソフトウェア配布には、一般に暗号が使用されるため、ネットワーク上の通信を盗聴したり CD-ROM からコピーしてファイルを入手しても、不正使用は不可能である。不正使用の可能性は、復号されたファイル／メモリ上のソフトウェアが、課金処理を経ずに繰り返し使用されることである。この不正使用を防止するための方法の一つとして、ファイル／メモリ上に暗号化されていないソフトウェアを置かない、即ちファイル／メモリ上のソフトウェアは常時１箇所以上暗号化されていることを保証する方法が考えられる。

A Study of Software Distribution
Toshikiyo Tanaka
NTT Information and Communication System Laboratories

図２に本方法によるソフトウェアの構成図を、以下に本方法による処理手順を示す。

① Decrypt()関数を呼び出し、流通クライアントに領域Aの復号を依頼する。

a) 流通クライアントは、課金サーバーに秘密鍵を要求する。

b) 課金サーバーは、ユーザ対応の課金情報を更新し、流通クライアントに秘密鍵を送付する。

c) 流通クライアントは、秘密鍵を用いて領域Aを復号する。

② 領域Aの命令を実行する。

③ Encrypt()関数を呼び出し、流通クライアントに領域Aの暗号化を依頼する。

a) 流通クライアントは、①b)の秘密鍵を用いて領域Aを暗号化し、秘密鍵を削除する。

（④～⑥については、領域Bに対して①～③と同様の処理を実行する。）

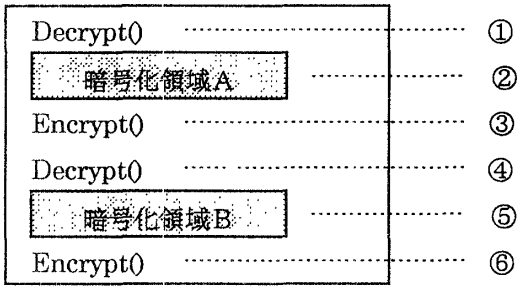

図２　ソフトウェア構成図

４．検討項目

上記不正使用防止方法を実現する上での検討項目と検討状況を以下に示す。

(1) 暗号化領域の数・・・図２の例では、暗号化領域の数は２であるが、２以上の任意の数とすることができる。ただし、暗号化領域の数が増えると流通クライアントと課金サーバーとの間の通信時間オーバヘッドが増えるため、ソフトウェア実行時間と不正使用攻撃に対する耐性とのトレードオフとなる。また、ソフトウェアの機能毎に暗号化領域を設定することにより機能単位の使用量課金が可能になる。

(2) 暗号化／復号関数の埋め込み・・・暗号化領域が復号されていない期間に暗号化領域へのアクセスや分岐が発生しないように、暗号化領域の設定および復号関数 Decrypt()、暗号化関数 Encrypt()の埋め込みは、ソフトウェア開発時に行わなければならない。開発済みのソフトウェアのソースコードを分析して、自動的に暗号化／復号関数を埋め込む方法については、今後の検討課題である。

(3) 売り切り方式とのソースコード共用化・・・暗号化／復号関数を埋め込んだソースコードをコンパイル・リンクし、指定領域を暗号化することにより、使用量課金方式に対応したソフトウェアを作成する。同一のソースコードをコンパイルし、内部処理をNOP 化したダミーの暗号化／復号関数をリンクすることにより、売り切り方式に対応したソフトウェアを作成でき、ソースコードの共用化が可能である。

(4) ソフトウェア実行時間の高速化・・・本方法では、ソフトウェア実行中に流通クライアントと課金サーバーとの間で通信を行うため、実行時間に対する通信時間オーバヘッドが存在し、またソフトウェア実行時に通信が可能でなければならないという制約条件がある。これらの問題への対処方法としては、課金サーバーとの通信で取得した秘密鍵を流通クライアントで保持し、ソフトウェア使用量を流通クライアントで累積して課金サーバーへ通知する方法等が考えられる。

５．おわりに

ソフトウェアの使用量に基づく課金について、専用ハードウェアを用いることなく、ソフトウェアのみで実現可能な方法を提案した。本方法では、ソフトウェアの複数の領域を暗号化し、ソフトウェア実行中のどの時点を取っても１つ以上の領域が暗号化されているため、ソフトウェア不正使用の攻撃に対してある程度の耐性を持つ。今後は、上記の実行時間高速化方法について更に検討を進めたい。

［参考文献］

１．森亮一、田代秀一：ソフトウェア・サービス・システム（SSS）の提案、電子情報通信学会論文誌、Vol.J70-D、No.1、pp.70-81(1987)