6 F － 8

Java用Publish/SubscribeミドルウェアSecure Distributed InfoBusにおける鍵配送プロトコル

丸山宏(日本アイビーエム東京基礎研究所及び東京工業大学情報理工学研究科)
maruyama@jp.ibm.com, maruyama@cs.titech.ac.jp
浦本直彦(日本アイビーエム東京基礎研究所)
uramoto@jp.ibm.com

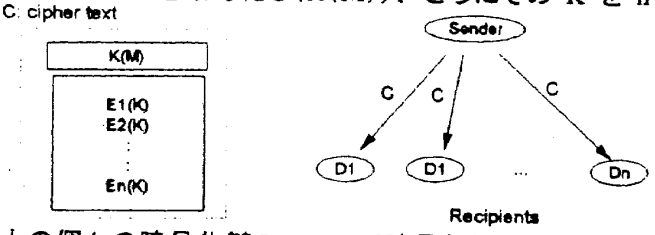
1. はじめに

衛星放送あるいは、インターネットにおけるマルチキャストのような放送型メディアにおいて、ユーザーの認証及びデータの暗号化を行う場合、受信者(subscriber)に鍵の配信を行わなければならない。マルチキャストを用いたJava用Publish/SubscribeミドルウェアSecure Distributed InfoBus[1]もこのような鍵配信を行わなければならない例である。新たに受信者が参加する場合には、その人に現在使っている鍵を渡せば良いが、受信者が参加をやめた場合、データの暗号鍵を更新する必要がある。本論文では、暗号鍵を木構造に構造化することで、暗号鍵更新のための手間を最小にする方法を提案する。

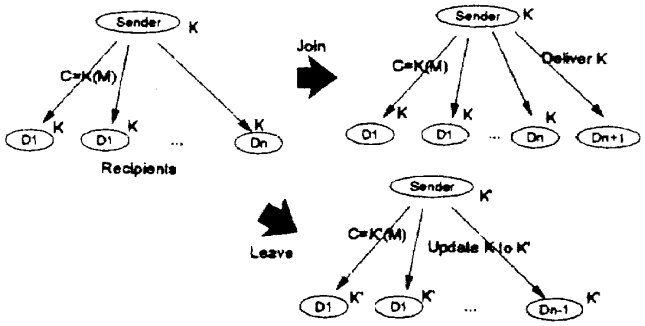
2. 解こうとする問題

受信者(subscriber)の集合をSとする。Sは例えば、特定の有料番組を視聴している視聴者である。あるいは、企業内で、ある機密の連絡を待っている従業員の集合かもしれない。インターネット上でよく使われているメイリングリストも、参加者の集合をSとするとこの例である。

Sにはn人の受信者がいるとする。放送者(publisher)Pはコンテンツcを、暗号化して送り出す。インターネットの世界では、標準化団体IETFが、メイリングリストの暗号化の標準化案として、RFC1421を提案している。これは、共通のセッション鍵Kでメッセージ Mを暗号化し(K(M))、さらにそのKをn人の個々の暗号化鍵D1...Dnで暗号化し(D1(K), D2(K), ..., Dn(K))、それをK(M)とともに、送り出すというものである。受信者は対応する復号化鍵(E1, E2, ..., En)でセッション鍵Kを複合化し、得られたKでメッセージ Mを復号化する。この方式はしかし、受信者の数nが大きくなると、メッセージに比して鍵パケットの大きさが非常に大きくなってしまうという欠点がある。例えば個人鍵に512ビットのRSA鍵を用いた場合、一つのDi(K)は最低64バイトになり、nが例えば 10,000だと一つのメッセージにつき、640KBytesの鍵パケットを送り出さなければならない。

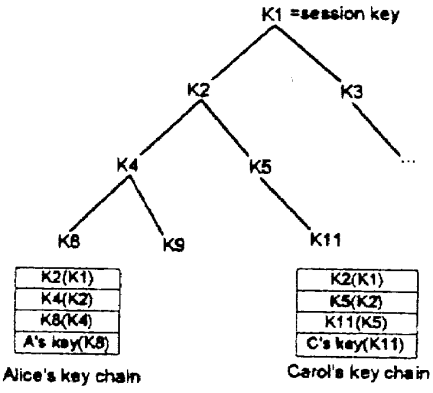


これに対して、我々は共通のセッション鍵Kをあらかじめ各受信者の個人鍵を使って配布しておく方法(Join/Leave Model)を考える。個人鍵は、一般的な公開鍵証明書に基づくものでもよいし、アプリケーション毎に用意されたものであってもよい。ここで問題となるのは、受信者の集合 Sに変更があった場合である。すなわち、Sに新たな要素が加わった場合、つまり新しい視聴者が参加した場合などは、そのユーザーの身元を個人鍵を使って認証し、K0を配れば良いが、Sの要素がSから抜けた場合には、鍵の更新が大変である。例えば、キャロルが支払日になっても視聴料を支払わなかったために、Sから除こうとしたとする。彼女はK0を持っているために、同じセッション鍵を使い続けるわけにはいかない。単純にやるとなると、新しい鍵K0'を生成し、それを残ったn-1人の受信者にそれぞれの個人鍵を使って送らなければならない。これでは、nが大きい場合、例えば10万人の視聴者がいる場合、一人が抜ける度に10万の鍵配送が起きてしまうことになる。以下に述べる方法によって、非常に小さいオーバーヘッドでセッション鍵の更新ができることを示す。



3. 提案する鍵分配方式

ここで提案する手法は、鍵をn進木の形に構造化する。まずは2進木の形で議論を勧める。図では、木のルートにあるK0がセッション鍵である。コンテンツはこの鍵で暗号化される。K0は、K1、K2でそれぞれ暗号化されて配送される。これをK1(K0)、K2(K0)と略記する。同様に、鍵K1はK3とK4で暗号化され、送信される。

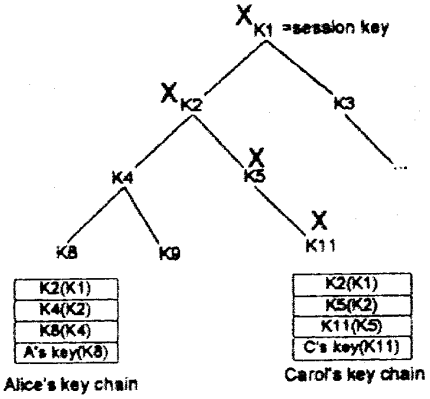


アリスがあるとき視聴料を支払って、正当な受信者となったとしよう。サーバーは、アリスに鍵K7を割り当て、アリスの公開鍵で暗号化して送る。また、K3をK7で暗号化したもの、すなわちK7(K3)、さらにK3(K1)、K1(K0)というように、ルートのセッション鍵に至る一連の鍵のチェーンをアリスに送る。受信者の全体数がnである場合、アリスのキーチェ

Key Distribution Protocol of SD-InfoBus, Hiroshi Maruyama (IBM Research, Tokyo Research Laboratory and Dept.of Computer Science, Tokyo Institute of Technology) and Naohiko Uramoto (IBM Research, Tokyo Research Laboartory)

ーンの大きさは log(n)ほどである。

さて、ここで、キャロルが何らかの理由で視聴者の集合 S から脱退したとしよう。キャロルは、彼女のキーチェーンに、K0、K1、K4、K10 という鍵を持っている。したがって、今後キャロルにコンテンツをアクセスさせないためには、これらの鍵を再発行して、キャロルの鍵を無効にしなければならない(下図で、X のついている鍵である)。



K1 を再発行して K1'としたとする。すると、アリスは自分の持っている K1 が無効になってしまうので、新しい K1' を K3 で暗号化した K3(K1')を受け取る必要がある。同様に、K0' は、K1'(K0') と K2(K0')という二つの鍵配送パケットで受信者に知らせる必要がある。これにより、アリスは K1'と K0' を知ることになり、次から新しいセッション鍵 K0'で暗号化されるコンテンツを見ることができるのである。

4. 鍵再配送の効率

キャロルが脱退したときの、鍵の再配送に必要なパケット数はいかほどだろうか？キャロルが持っていた鍵は $\log_2(n)$個であるから、新しく生成しなければならない鍵の数も $\log_2(n)$である(厳密にいうと上の方式では $\log_2(n)$より1小さい)。新しい鍵 1 個について、その鍵を、その 2 つの子供の鍵で暗号化して送る必要がある。従って、鍵の再配送に必要な鍵配送パケットの数は、$2*\log_2(n)$である。一般に r 進木の場合を考えると、鍵再配送パケットの数 p は $p = r*\log_r(n) = r*\log(n)/\log(r)$となる。

n が一定の時、p を最小にする r は、r=e である(e は自然対数の底)が、実際には r は自然数であるので、r=3 の時が最適となり、パケット数はおよそ $2.73*\log(n)$となる。

例えば、n が 100 万、すなわち、100 万人の視聴者にコンテンツを放送しているとしよう。ある一人の視聴者を外すためには、2 進木の場合 $2*\log_2(10^6)$個すなわち、40 個の鍵再配送パケットを放送すればよい。3 進木の場合は $2.73*\log(10^6)=37.7$ であるので、鍵の暗号化方式として DES を用いるとすれば、一つの鍵を 64 ビット+鍵の ID で送れるので、鍵の ID を 32 ビットとしても、鍵配送パケットのペイロードは 96 ビット(12 バイト)であり、これが 38 個で全体が 0.5K バイトに収まる。

5. 鍵チェーンの再計算

キャロルが脱退したとき、アリスは新しい鍵配送パケットを受け取って、自分のキーチェーンを更新しなければならない。キャロルが鍵の鍵構造木のどこにいたにせよ、K0 は必ず更新されるから、K0'の計算はいずれにせよやる必要がある。K1'についてはどうだろうか？キャロルがアリスと同じ枝にいる確率は、(2 進木の場合、木がバランスしてるならば)0.5 である。したがって、K1 を更新しなければならない確率は 0.5 である。同様に、K3 を更新しなければならない確率は 0.25 になる。n が非常に大きくなったとしても、アリスが更新しなければならない鍵の数の期待値は、$1+0.5+0.25+0.125+ \ldots. = 2$ を越えない。つまり、鍵チェーンの再計算に必要な復号オペレーションの期待値は 2 進木の場合2である。同様に、r 進木の場合、鍵再計算数の期待値の上限は r/(r-1)である。

6. 複数の脱退をまとめて計算する場合

もし、k人がまとめて脱退した場合、$k*\log_2(n)$個の鍵を新たに生成する必要はない。例えば、K0 は 1 回だけ更新すればよいからである。また、何人かが同時期に脱退することがあらかじめわかっている場合(月末までの契約で視聴している場合、など)は、それらの視聴者をできるだけ同じ枝にまとめることで、複数の脱退が起きたときの鍵の更新と再配送を小さくすることができる。

7. 既存技術との比較

有料衛星通信などに現在使われている技術は、秘密アルゴリズム方式で各社の方式に互換性がなく、また、デコーダの所有が鍵の所有となっているので、ダイナミックに新たな視聴者を参加させたり一時的に脱退させたりすることができない。ペイパービューなどで使われている方式は、鍵の配送を電話による1対1の通信に頼っており、上り回線を必要とする、スケールしない、などの問題点がある。我々の方式は、既知の暗号技術の上で動くために規格を公開でき、各社の相互接続性が保たれ、また、上り回線を使うことなく、ダイナミックに特定の視聴者を参加させたり脱退させたりすることができる。鍵の配信も放送で行われるために、スケールが大きくなっても、鍵の配信はボトルネックになりにくい。

インターネットでのマルチキャストのセキュリティとして、ルータからグループ鍵を配信する方法が提案されている。しかし、この方式では、信頼できるルータをネットワーク上にあらかじめインストールしておかなければならない、という重大なセキュリティ上の問題がある。また、この方式は、IP のレイヤーでのセキュリティであり、アプリケーションからのエンドツーエンドのセキュリティを保証するものではない。我々の方式では、エンドツーエンドの方式であり、途中にいくらセキュリティ上の問題がある経路があっても全体のセキュリティは保たれるのである。

参考文献

浦本，丸山 「Java用Publish/Subscribeミドルウェア Secure Distributed InfoBusの設計と実装」、情報処理学会56回全国大会予稿集

Bruce Schneier, "Applied Cryptography," John Wiley & Sons, New York, 1996.

K. Matsumura, Y. Zheng, and H. Imai, "Analysis ,of and Improvements on CBT Multicast Key-Distribution," (*working draft*), July 29, 1197.

Pekka Pessi, "Secure Multicast," (http://www.nixu.fi/~pnr/netsec-lopulliset/3-0-multicast.html), 1996.

IETF, "IP Authentication using Keyed MD5", RFC1828, (http://ds.internic.net/rfc/rfc1828.txt), 1995.

IETF, "Privacy Enhancement for Internet Electronic Mail", RFC1421, (http://ds.internic.net/rfc/rfc1421.txt), 1993.

Hugh Harney, Carl Muckenhirn, Thomas Rivers. Group Key Management Protocol (GKMP) Architecture. (I-D draft-harney-gkmp-arch-00.txt), September 1994

Phil Karn, William A Simpso, "The Photuris Session Key Management Protocol", IETF working draft, http://www.tcm.hut.fi/Opinnot/Tik-110.501/1995/draft-ietf-ipsec-photuris-06.txt, 1995.