輪講資料 04/9/15

Sharad Goel, Mark Robson et al.,

“Herbivore: A Scalable and Efficient Protocol for Anonymous Communication”, Cornell University Computing and Information Science Technical Report, TR2003-1890, February 2003.

資料作成: 門林 雄基（奈良先端科学技術大学院大学）

1 Abstract

匿名性を取り扱う。peer-to-peer, スケーラブル、耐タンパな通信システムとして Herbivore を提案。証明可能 (provable) な匿名性，プライバシを提供。“Dining cryptographer networks” (DC-nets) に基づき構築。効率とスケーラビリティを実現しており、この点が他の匿名通信プロトコルと異なる (と主張)。

2 DC-nets background

DC-net で 1bit 送る場合。Alice, Bob が Charlie にメッセージを送る。Alice は coin toss の結果をそのまま伝える。Bob は coin toss の結果と送りたいビットを XOR して送る。Charlie は Alice, Bob からもらったビットを XOR すればメッセージが得られる。このとき Alice, Bob のいづれがメッセージを送ったかはわからない。

数学的にいえば、coin toss というのは Alice, Bob が共有している乱数列だといえる。任意の乱数 r について $r \oplus r = 0$ なので、$(x \oplus r) \oplus (0 \oplus r) = x$ である。

より正確な数学的取り扱いは p.17 Appendix A を参照。

3 Protocol

Round protocol: 参加ノード間でのビットの送り方を規定する。参加ノード数が増えると効率が悪くなってしまうので、分割統治する。

Global topology control algorithm: ネットワークを小さな匿名クリーク (anonymizing clique) に分割する (Figure 1 参照)。それぞれのクリークには少なくとも k ノードが存在する。k は匿名性の度合いを決める定数。

3.1 Random entry protocol

k ノードからなる匿名クリークを生成・維持・解消するためのアルゴリズムについて述べている。

まず複数の k ノードからあるクリークがあるとして、狙ったクリークに join できてしまうと攻撃者に囲まれてしまい匿名性が失われてしまう可能性がある。このためランダムなクリークにしか join できないような仕組みを考える。

IP アドレスをハッシュしたものや (IP, salt) をハッシュしたものをノード ID (DHT 上の) とする先行研究が目立つが、これだと IP アドレスを無限にもつ攻撃者にたいして弱い。そこで以下のような方法を考える。

まずノードは join するまえに公開鍵と秘密鍵の対 $K_{public}/K_{private}$ を生成する。下位 m_k ビッ

トについて $f(K_{public}) = f(y)$ となる y をみつけるまで乱数を生成する。$g(K_{public}, y)$ がノード ID。ここで f, g は適当なハッシュ関数。ほかのノードは (K_{public}, y) からノード ID の正当性を検証できる。詳細略。

このような y を発見することは m_k が大きいと容易でないので攻撃者への抑止力があると考えられる。

このほか y をあらかじめ計算しておく辞書攻撃への対処として下位 m_k ビットの一部に時刻をつかう方法が提案されている。

3.2 Global topology control

クリークの大きさを k から $3k$ までの間に保つ方法は以下の通り。まず大きさ $3k$ になったクリークを解消する。つぎにクリークの ID として K_{C_1}, K_{C_2} を計算し、これを Chord の key としてクリーク C_1, C_2 を生成する。詳しくは p.6 参照。直感的には Figure 2. （ここの記述は怪しい。Chord は range query をサポートしていないので、Chord に若干の拡張が必要なはず。）

3.3 補足

以下の議論ではいくつか前提があるのでここで補足する。

1. クリーク内の任意の 2 ノードはは互いに乱数系列を共有している。つまり key graph はフルメッシュである。
2. クリーク内のメッセージは、中心ノードを中心としてスター状に配送される。中心ノードは動的にえらばれる。
3. 中心ノードはほかのノードから送られてきたメッセージをすべて（乱数系列と XOR して）元に戻して、XOR してほかのノードにブロードキャストする。
4. クリーク内のすべてのノードは、1) メッセージを中心に送る、2) 中心ノードで XOR する、3) ほかのノードにブロードキャストする、という 3 ステップを同期しておこなう。

なお本論文ではクリーク内での通信については述べられているが、クリーク間の通信については記述が無い。これは DHT を使って実現できるためだと思われる。あるいはクリークを単一ノードとみなし、階層化することで “super clique” を構成することもできる。ただしこの場合、通信量が大きくなる。

3.4 Round protocol

Reservation phase: 乱数 i を $1 \ldots m_r$ のなかからえらび、m_r ビットベクタの i ビットを 1 にして匿名で送信する。これにより送る順序を決める。偶数ノードが衝突したときはわかる。奇数ノードが衝突したときはわからない。

Transmission phase: 送る順番がきたノードはメッセージを送る。ほかのノードは 0 を送る。匿名通信路を見れば、送信が成功したかどうかわかる。

Exit phase: 長い通信をクリーク上でおこないたいときに、他のノードがクリークから脱退しないようにしらせるためのもの。

Reservation phase でつかう m_r ビットベクタの長さはどれくらいが適当か。p. 9 に導出過程。$m_r = k_r\sqrt{p}$ ここで p はパケットサイズ、k_r は当該ラウンドに送信するノード数の期待値。

4 Attacks

Collusion は random entry protocol によって防げる。

Sybil attacks はノード新規加入に計算量のオーバーヘッドを課することである程度ふせぐことができる。

長いやりとりやメンバーシップの変化を監視することで Intersection attack が可能（つまり長いメッセージがやりとりされており、長い間メンバーであったノードが２つしかないのであれば送受信者が露呈する、という脆弱性）。これについては Exit protocol で対処。

統計的分析による攻撃も考えられる。

Coordinator attack: 中央ノードが乗っ取られた場合の脆弱性について。1 out of every k rounds なので影響は限定できるとしている。

Exit attack 略

Denial of Service: クリークが攻撃された場合、異なるクリークに加入すればよい。（浅い考察）

5 Performance Evaluation

PlanetLab を用いた評価。クリークの大きさを１０から４０に変化させた。

Figure 5: 匿名通信の帯域幅。送信者数１～４。詳細説明なし。

Figure 6: 匿名通信の遅延。送信者数１～４。詳細説明なし。

Figure 7: 同時に送信したいノードが複数いる場合の帯域利用率。ここで帯域利用率とは、送信した匿名ビット数 / 送信したビット数。

Figure 8: 固定長のバイト列を送るためにノードが送らなければならないビット数。