2008－DPS－135　(18)
2008／6／20

SRJE: 結託攻撃を抑制するための分散認証手法

真下 洋† 重野 寛†
慶應義塾大学大学院理工学研究科†

P2P ネットワークにおいて，悪意あるユーザが複数ノードを用いてネットワークを支配しようとする結託攻撃 (Sybil Attack) がある．結託攻撃の抑制には，ユーザごとのノード数を制限することが有効であり，DHT ネットワーク上でのノード数制限を実現するための分散認証手法 Self-Registration が提案されている．しかし Self-Registration には，悪意あるユーザの増加に耐えられないという問題がある．本稿では，Self-Registration に独自の監視システムを用いたローカルレピュテーションを適用することで信頼性を高めた分散認証手法 SRJE を提案する．シミュレーションを用いて Self-Registration と SRJE を比較評価し，SRJE の有効性を示す．

SRJE: Distributed Authentication Method to Inhibit Sybil Attacks

Yo MASHIMO† Hiroshi SHIGENO†
Graduate School of Science and Technology, Keio University†

In P2P networks, there are the attacks malicious users try to control the networks by using multiple nodes, called Sybil Attacks. To inhibit Sybil Attacks, it is thought that the restriction of the number of nodes per user is effective. To realize the restriction on DHT networks, a distributed authentication method Self-Registration was proposed. However, Self-Registraion is not tolerant of increase of malicious users. In this paper, we propose SRJE which is a distributed authentication method applied a local reputation using an original evaluation system for Self-Registration. And then we compare SRJE with Self-Registration through computer simulation.

1 はじめに

近年，構造化 P2P の研究が盛んになってきている．P2P には非構造化 P2P と構造化 P2P があり，非構造化 P2P を利用したソフトウェアとして Gnutella や WinMX などが知られている．しかし，非構造化 P2P ネットワークでは，フラッディングを用いたキー探索を行うため，トラフィックの圧迫や探索ホップ数の増加といった，スケーラビリティに関する問題がある．そこで，スケーラビリティを考慮した構造化 P2P，特に分散ハッシュテーブル (DHT) を利用した Chord[1]，Pastry[2]，CAN[3]，Kademlia[4] といったアルゴリズムが提案されてきた．最近では，BitTorrent[5] などのソフトウェアにも応用され始め，広く知られるようになってきた．

一方，P2P ネットワークでは悪意あるノードをネットワーク上に配置することが容易であり，セキュリティに関する問題点も数多く挙げられている．悪意あるノードがネットワークに入り込んだ際の被害を軽減するため，ネットワークの頑健性を高める研究も多くなされているが，悪意あるノードがネットワークに多数入り込んでしまった場合，それらによってネットワーク全体が機能不全に陥ってしまう事が知られている．このように，悪意あるユーザが多数のノードを利用してネットワークの支配を図る攻撃を結託攻撃 (Sybil Attack[6]) という．

これに対し，分散的にユーザの参加の可否を決定する，SybilGuard[7]，Geometric Certification Protocols[8]，Self-Registration[9] といった分散認証手法が提案されている．特に Self-Registration は，1ユーザが持つノードの数を複数のノードで把握し，多数決による参加許可によってユーザが一定数以上のノードを参加させることができない様にする．

Self-Registration は，悪意あるユーザの割合が小さい場合には効果を発揮するが，多数決を用いているため，悪意あるユーザの割合の増加に耐えられない問題がある．

本稿では，Self-Registraion の問題点を，ローカルレピュテーションを用いることによって改善した認証手法 Self-Registration with Judgement Evaluation(SRJE) を提案し，シミュレーションによって Self-Registration との比較評価を行う．

以下，本稿では，第2章において Self-Registration 及び SRJE の関連研究について説明する．第3章において，Self-Registration の動作と問題点について述べる．第4章では SRJE を提案し，第5章でシミュレーションによる Self-Registration との比較評価を行う．第6章をまとめとする．

2 関連研究

本章では関連研究として，DHT のアルゴリズムの 1 つである Chord[1]，結託攻撃 [6]，及び Self-Registration[9] について説明する．

2.1 Chord

Chord は代表的な DHT の 1 つである．Chord ネットワーク上では，各ノードは m ビットの整数値の ID を持ち，IP アドレスとポート番号を変換したハッシュ値を使うことが想定されている．また，コンテンツについても，同様の ID が付与され，その ID に最も近く，かつ大きい ID を持つノードに所有される仕組みになっている．

Chord ネットワーク上のノードは，リング状のトポロジを持ち，ネットワーク内での検索クエリはこのリング上で ID の増加する方向に転送される．各ノードはコンテンツの探索クエリを転送するためのルーティングテーブルを持ち，そのエントリ数は m 個となっている．ID が ID_i であるノードの検索クエリ転送先のノード ID は以下で与えられる．

$$ID_j = ID_i + 2^j. \tag{1}$$

式 (1) により求められる ID を持つノードは存在しない可能性があるが，その場合は，その ID に最も近く，かつ大きい ID を持つノードがクエリ転送先となる．この実際にクエリが転送されるノードを successor と呼ぶ．

また，あるノードから見て，その ID に最も近く，かつ小さい ID を持つノードの事を predecessor という．新規ノードの Chord ネットワークへの参加は，新規ノードの predecessor が自身のルーティングテーブルに新規ノードを加えることによって行われる．

2.2 結託攻撃

文献 [6] では，悪意あるユーザが複数のネットワーク上の ID を持つことができる時，最終的にそのユーザがネットワーク全体を制御できるようになることを示している．具体的には，クエリの転送を妨害したり，検索クエリの改竄，クエリのフラッディングなどを行うことによって，ネットワークを機能不全に陥らせる事ができる．さらにこの文献では，ある限定的な条件がない限り，信頼できる第三者認証機関を設置することが，結託攻撃を抑止する唯一の方法であると述べている．

しかし，中央機関を設置することは，スケーラビリティや耐障害性といった P2P の特長を損ないかねない．そのため，結託攻撃への対抗手法として，中央機関を用いずに分散的に認証を行う技術の研究が行われている．

文献 [7] では，ソーシャルネットワークにおけるネットワークトポロジの性質を利用した認証方式 SybilGuard を提案している．ソーシャルネットワークにおいては悪意あるユーザが持つ複数のノードは，他のユーザが持つノードとのリンクが少ないことが知られており，SybilGuard ではそれを検知することで同じユーザが作り出したかどうかの判別とノード認証を行う．

文献 [8] では，同一のユーザが操作するノードを特定できる P2P ネットワーク，Geometric Certification Protocols を提案している．Geometric Certification Protocols では，悪意ある参加者が操作するノードは物理的に近い距離にあるとみなし，RTT（Round Trip Time）によって悪意あるノードの所在のまとまりを検知する．ここで，RTT がほぼ等しいノードは同じユーザが操作するものとみなしており，悪意あるノード群の検知もこれに基づくものである．

3 Self-Registration

本章では，結託攻撃を抑制する分散認証手法の一種である Self-Registration[9] について説明し，その問題点を指摘する．

3.1 マルチ ID

1 ユーザが a 個のノードを扱えるマルチ ID 環境において，悪意あるユーザの割合を p_u とした時，悪意あるノードの割合 p_n は，

$$p_n = \frac{N_{mal}}{N} = \frac{a \times p_u}{(1-p_u) + a \times p_u} \tag{2}$$

(N_{mal} : 悪意あるノード数, N : 全ノード数)

と表すことができる．ここで，正常なユーザは 1 ノード，悪意あるユーザは最大の a ノードを持つと仮定している．

図 1 は，式 (2) の関係をグラフにしたものである．この図から，p_u が小さい時，a の増加と $\frac{\Delta p_n}{\Delta p_u}$ の増加が同値であることが判る．従って，a を制限することが結託攻撃を抑制する一つの手段になり得る事が推測される．この a の制限を実現するために，Self-Registration が提案された．

3.2 Self-Registration の動作

Self-Registration は，1 ユーザが持つノード情報を複数の認証ノードで分散的に管理し，そのユーザ

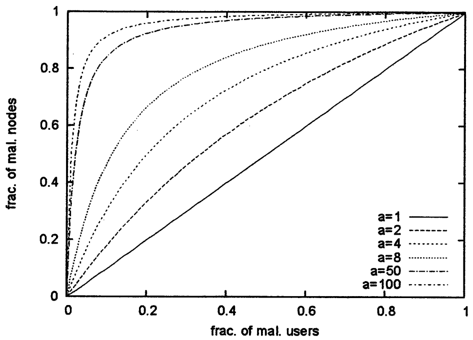

図 1: マルチ ID 環境における p_u と p_n の関係

が新たにノードを持とうとした時に制限数の確認を行い，多数決によって認証を行うものである．

まず，新規ノード n_{new} は自分自身の ID id_{new} を (3) 式に従って算出する．

$$id_{new} = hash(IP_{new} \oplus port_{new}). \quad (3)$$

ここで，$\oplus$ は String の連結を表す．

次に，自分のノード情報を登録する r 個の認証ノードを決定し，それらに predecessor n_{pre} を通じて参加クエリを送る．j 番目 $(1 \leq j \leq r)$ の認証ノード ID$regId_i^{new}$ は，で決定される．

$$regId_i^{new} = hash(i \oplus IP_{new}). \quad (4)$$

$regId_j^{new}$ がノード (IP+port) 毎ではなくユーザ (IP) 毎に決定されるため，認証ノードはユーザの持つノードを管理することができる．(4) 式によって決定される認証ノード群を本稿では認証グループと呼ぶ．

参加クエリを受け取った認証ノードは，n_{new} の ID の真正性と制限ノード数 (a) 非超過の確認を行う．この 2 つの確認から，新規ノードの参加の可否を判断し，他の認証ノードと判断を交換する．可の判断が半数を超えた時，n_{new} は n_{pre} によってネットワークに加えられる．

3.3 Self-Registration の問題点

悪意あるノードは，認証されるべきでないノードの認証と，認証されるべきノードの認証失敗を図る．これら 2 つの事象を合わせて誤認証と呼ぶ．具体的には，悪意あるノードは誤った判断を他の認証ノードに送る．故に，Self-Registration において，誤認証は認証グループの半数以上を悪意あるノードが占めると発生する．

悪意あるノードの割合 p_n，認証ノード数 r の時，悪意あるノードが認証グループ内に k ノード存在する確率 $g(r,k,p_n)$，及び誤認証の発生確率 $h(r,p_n)$ は以下のように表せる．

$$g(r,k,p_n) = (1-p_n)^{r-k} \times p_n^k \times {}_rC_k. \quad (5)$$

$$h(r,p_n) = \sum_{i=\lceil \frac{r}{2} \rceil}^{r} g(r,i,p_n). \quad (6)$$

$r=5$，$a=2$ の環境で，$p_u = 2\%$ の時，$h(r,p_n) = 0.06\%$ である．

この様に，p_u が小さい場合には誤認証の発生確率は小さく，Self-Registration が有効に動作する．しかし，p_u が大きくなると誤認証の発生確率が大きくなる．先の環境において，$p_u = 10\%$ では $h(r,p_n) = 4.49\%$ まで上がってしまう．従って Self-Registration は，悪意あるユーザの増加に耐えられない不安定なシステムである．

この問題に対し Dinger らは，認証ノード数 r を増やすことによって誤認証の発生を抑制することができると述べている．しかし，r の増加が判断交換によるオーバヘッドの増大 $(O(r^2))$ に繋がってしまう事も指摘している．そのため，r を増やさずに誤認証の発生を抑制する必要がある．

4 SRJE

本章では Self-Registration における前述の問題点をローカルレピュテーションを利用して改善した，Self-Registration with Judgment Evaluation (SRJE) を提案する．

4.1 SRJE の概要

第 3 章で示したように，Self-Registration は悪意あるユーザの増加に耐えられない．その原因は，悪意あるノードが誤った判断を他の認証ノードに送り，その判断が多数を占めてしまった時に起こる誤認証にあった．そこで，誤った判断を他の認証ノードに送ったノードの評価値を下げておくことによって，次回以降の判断交換におけるそのノードの判断を軽視する手法を Self-Regsitration に組み込んだ，新しい分散認証手法 Self-Registration with Judgment Evaluation (SRJE) を提案する．

ノードの評価方法として，SRJE では以下の 3 つの監視を採用した．

1. 認証ノードによる認証ノードに対する監視
2. predecessor よる認証ノードに対する監視
3. 認証ノードによる predecessor に対する監視

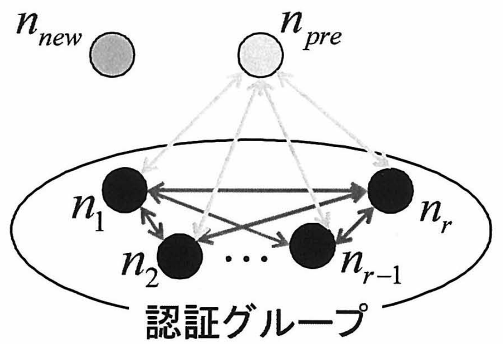

図 2: ノード評価の概念図

図2は3つの監視を用いたノード評価の概念図である．これらの監視を用いて得た評価値を，それぞれがローカルレピュテーションとして自身の持つ評価値テーブルに蓄えていき，次回以降の認証で利用する．評価値テーブルは，表4.1に示されるようなノードIDをキーとするテーブルで表される．

表 1: 評価値テーブル

ノード	評価値
id_2	3
id_4	0
id_{31}	2

4.2 SRJEの動作

図3はSRJEの動作の概要を表す．以下では，新規ノードn_{new}，認証ノードn_i，predecessorn_{pre}それぞれの動作と，評価値テーブルの更新方法について説明する．

4.2.1 n_{new} の動作

n_{new}は，Self-Registrationと全く同じ動作をする．

まずn_{new}は，自身のIDid_{new}を式(3)に従って算出し，次に各認証ノードのID $regId^i_{new}$ $(1 \leq i \leq r)$を式(4)によって決定する．最後に$regId_{new}$をキーとする参加クエリを送信する．

4.2.2 n_i の動作

参加クエリを受け取ったn_iは，認証ノードとして動作する．

まず，n_{new}のID id_{new}の真正性とノード数制限非超過の確認を行う．具体的には，pingを用いて確認されたIPアドレスとポートから式(3)を用いて算出したid_{new}と，表2に示すユーザ-ノードリストマップを使用する．以上の確認で異常がなければ

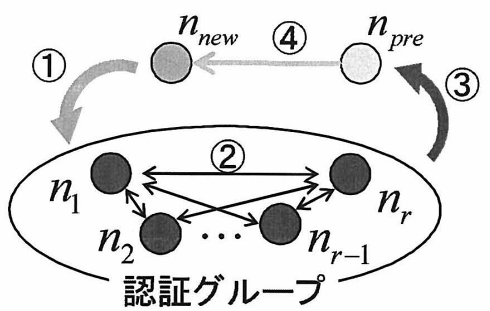

図 3: SRJE の動作

表 2: ユーザ-ノードリストマップ

ユーザ	ノードリスト
IP_1	n_1
IP_2	n_{21}, n_{22}
IP_3	n_3

$true$を，あれば$false$を自身の判断j_iとする．

(4)式から求めた他の認証ノードと判断の交換を行い，集めた判断を$\boldsymbol{J} = [j_j \,|\, 1 \leq j \leq r]$としてベクトル化する．この際，$\boldsymbol{J}$を計算に使用するため，

$$j_j = \begin{cases} 1 & (true) \\ 0 & (timeout) \\ -1 & (false) \end{cases} \quad (7)$$

として，整数に変換する．自身の評価値テーブルから，判断交換を行った認証ノードの評価値ベクトル$\boldsymbol{E} = [e_j | 1 \leq j \leq r]$を得る．ここで，対象ノードが自身の場合は決められた値，評価値テーブルのキーにある場合はその値，その他の場合はデフォルトの値を評価値e_jとする．

$\boldsymbol{J}$と$\boldsymbol{E}$の内積から，自身の決断d_iを決定する．

$$d_i = \boldsymbol{J} \cdot \boldsymbol{E}. \quad (8)$$

d_iをその正負によって$true/false$に変換し，担当者n_{pre}に送信する．$d_i = 0$の場合は$d_i = j_i$とする．$d_i = true$であった場合，参加者-ノードリストマップにこのn_{new}を加える．

最後にn_{pre}からd_{pre}を受け取り，評価値テーブルを更新する．

4.2.3 n_{pre} の動作

各認証ノードから受け取った決断d_j $(1 \leq j \leq r)$を，式(7)と同様に変換し，$\boldsymbol{D} = [d_j \,|\, 1 \leq j \leq r]$と

評価値ベクトル $\boldsymbol{E}$ との内積により，d_{pre} を求める．

$$d_{pre} = \boldsymbol{D} \cdot \boldsymbol{E}. \tag{9}$$

d_{pre} をその正負によって $true/false$ に変換する．$d_{pre} = true$ ならば n_{new} の参加を認め，自身のルーティングテーブルに加える．$d_{pre} = 0$ の場合は false negative 防止のために $d_{pre} = false$ とする．false positive であった場合，制限ノード数が許せばユーザはポート番号を変えて再試行することになる．

最後に，評価値テーブルの更新のため各認証ノードに d_{pre} を送り，自身の評価値テーブルも更新する．

4.2.4 評価値テーブルの更新方法

認証ノードの評価値テーブルは，最終結果，即ち d_{pre} を受け取った時に更新される．d_{pre} が自分の決断 d_i と異なっていた場合，n_{pre}が疑わしいとして，npre の評価値を下げる．そうでない場合は，ノード群 $\{n_j|d_j \neq d_{pre}\}$ の評価値を下げる．評価値テーブルのキーに n_j がない場合は追加を行う．これにより，3 つの監視のうち，認証ノードによる認証ノードに対する監視，認証ノードによる predeceossor に対する監視が実現できる．

predecessor の評価値テーブルも，d_{pre} を算出した後で更新される．更新は，ノード群 $\{n_j|d_j \neq d_{pre}\}$ の評価値を下げる事で行われる．評価値テーブルに該当するノードがない場合は追加を行う．predecessor の評価値テーブルの更新によって，predeceossor よる認証ノードに対する監視が実現される．

評価値は，時間経過に対して単調に減少する．これは，評価値を上昇させた後で悪意ある行為に及ぶ攻撃を防ぐためである．

5 評価

Self-Registration と SRJE を比較評価するため，シミュレーションを行った．

5.1 シミュレーション環境

シミュレーションモデルは以下の通りである．

1. 初期ノードを Chord ネットワークを想定したリストに配置し，一定時間おきにユーザがノードの参加を試みる
2. ユーザには正常なユーザと異常なユーザがいる
3. 正常なユーザはネットワークへの参加試行を 1 回だけ行う
4. 異常なユーザは全て悪意あるユーザである

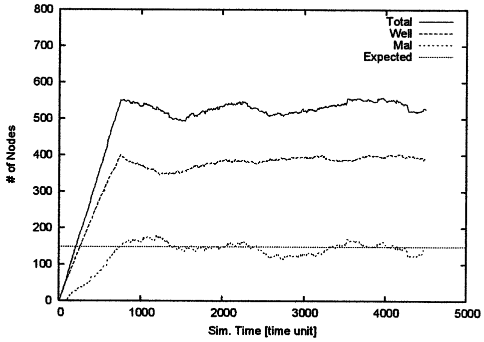

図 4: 悪意あるノード数の変化 (SRJE)

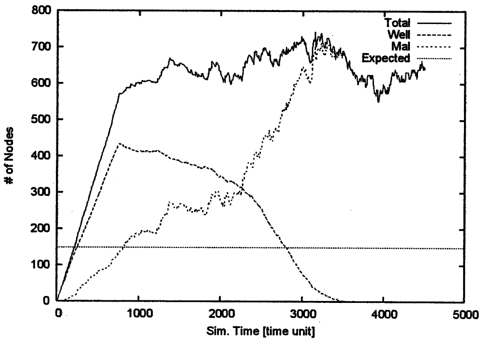

図 5: 悪意あるノード数の変化 (Self-Registration)

5. 悪意あるユーザは，20 ノードの参加を試み，参加後は誤認証の誘発を図る（則ち，判断を偽る）
6. 故障により誤判断をするノードは悪意あるノードと区別しない
7. 転送妨害攻撃，パケットロスなどによるメッセージロスは考慮しない

シミュレーションパラメータを表 5.1 に示す．以上の環境で，悪意あるユーザの割合が 15% の時の経過時間に対するノード数の変化と，悪意あるユーザの割合の変化に対するノード数制限成功率を測定

表 3: シミュレーション条件

1 参加者当たりの制限ノード数 a	2
認証ノード数 r	10
ユーザ参加間隔	15 time unit
ノードのライフタイム	7500 time unit
デフォルト評価値	10
自身に対する評価値	20

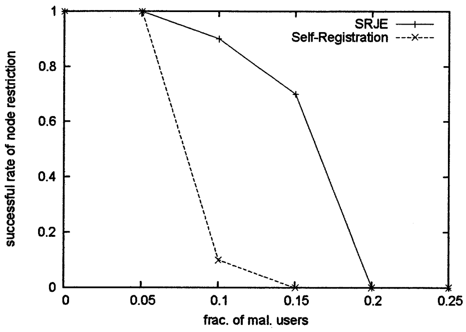

図 6: ノード数制限の成功率

した．ここで，成功とは十分な時間を行った結果悪意あるノードの割合が全体の半数を超えないことを示し，10 回の試行に対してノード数制限の成功した割合を成功率とした．

5.2 考察

Self-Registration でのシミュレーション結果を図 5 に，SRJE での結果を図 4 に示す．グラフは横軸がシミュレーション時間で，縦軸がノード数である．Total，Well，Mal はそれぞれ，ネットワーク全体のノード数，正常なノード数，悪意あるノード数を示し，Expected は式 (2) から導かれた期待値を示す．

図 5 及び図 4 に示すように，Self-Registration では悪意あるノードがネットワークを支配してしまった場合でも SRJE では悪意あるノード数を期待された値に抑えられていることがわかる．

図 6 は Self-Registration 及び SRJE におけるノード制限成功率を示す．図 6 から，常に SRJE の成功率が Self-Registration 以上の値を示し，最大で 80% の差をつけていることがわかる．悪意あるユーザの割合が Self-Registration では 10%，SRJE でも 20% 以上になると，成功率が急激に低下している．これは誤認証の発生が急激に増加し，悪意あるノードの増加と誤認証発生のスパイラルが起きているためと考えられる．

6 おわりに

本稿では，P2P ネットワークにおける攻撃の一種である結託攻撃を抑制する分散認証手法 Self-Registration に着目し，悪意あるユーザの割合の増加に耐えられないという問題点を示した．認証ノードと predecessor の相互監視によるローカルレピュテーションを用いた，より信頼性の高い認証手法 SRJE を提案し，その効果についてシミュレーションによる Self-Registration との比較評価を行った．比較評価の結果，悪意あるユーザの割合が大きく，Self-Registration では抑制しえなかった場合でも SRJE では抑制できることが判った．また，ノード数制限の成功率も，常に Self-Registration 以上の値を維持しており，最大で 80% の優位を見ることができた．以上から，SRJE の有効性を確認することができた．

謝辞

本研究の一部はグローバル COE プログラム「アクセス空間支援基盤技術の高度国際連携」により行われました．

参考文献

[1] I.Stoica, R.Morris, D.Nowell, D.Karger, M.Kaashoek, F.Dabek, and H.Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. *IEEE/ACM Transactions on Networking*, Vol. 11, No. 1, pp. 17–32, February 2003.

[2] A.Rowstron and P.Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pp. 329–350, 2001.

[3] S.Ratnasamy, P.Francis, M.Handley, R.Karp, and S.Shenker. A Scalable Content Addressable Network. In *ACM SIGCOMM*, pp. 161–172, 2001.

[4] P.Maymounkov and D.Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *Proc. the 1st International Workshop on Peer-to-Peer Systems (IPTPS),*, pp. 53–65, 2002.

[5] the bittorrent official web page. http://www.bittorrent.com/ 2008.1.15.

[6] J.Douceur. The Sybil Attack. In *Proc. the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, pp. 251–260, 2002.

[7] H.Yu, M.Kaminsky, P.Gibbons, and A.Flaxman. SybilGuard: Defending Against Sybil Attacks via Social Networks. In *ACM SIGCOMM*, pp. 267–278, September 2006.

[8] R. Bazzi and G. Konjevod. On the establishment of distinct identities in overlay networks. In *Proc. ACM symposium on Principles of distributed computing*, pp. 312–320, 2005.

[9] J.Dinger and H.Hartenstein. Defending the Sybil Attack in P2P Networks: Taxonomy, Challenges, and a Proposal for Self-Registration. In *Proc. the First International Conference on Availability, Reliability and Security (ARES'06)*, 2006.