匿名通信システム Tor の安全性を低下させるトラフィック逆加工

横手 健一　　松浦　幹太

東京大学生産技術研究所
東京都目黒区駒場 4-6-1 153-8505
{velleykt, kanta}@iis.u-tokyo.ac.jp

あらまし 匿名通信システム Tor はトラフィックに様々な加工を施した上で通信を行うため，トラフィックに着目した指紋攻撃に対してある程度の耐性を保持している．本研究ではこの加工を打ち消すような逆加工によって，Tor の安全性が更に低下することを示す．そして，この逆加工の潜在的な脅威について議論し，Tor が解決すべき今後の課題について考察する．

Anti-transform of Tor traffic to threaten its anonymity.

Ken-ichi Yokote　　Kanta Matsuura

Institute of Industrial Science, The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, JAPAN
{velleykt, kanta}@iis.u-tokyo.ac.jp

Abstract This paper proposes the Anti-transform of Tor traffic ,discusses the potential threat of it,and considers the solution to the future problem.

1　はじめに

近年インターネットは急速に普及し様々な用途で利用されるようになった. しかし, 手紙や電話の様な他の通信手段と異なり，インターネットはプライバシーに対する安全性が十分でない. 例えば，IT に精通していない者でも一般配布されているツールを用いることで容易に他者の行動を監視できてしまう．したがって，インターネット上におけるプライバシーの保護は重要な課題である．

暗号化通信は，情報を保護する手段の一つであるが，これだけで全てのプライバシー保護が達成できるわけではない．例えば，暗号化通信では通信の経路情報まで秘匿することはできない．そこで，通信経路を秘匿する目的で考案されたプライバシー保護技術 (Privacy Enhancing Technology) が，匿名通信システムである．匿名通信システムは David Chaum[2] によって 1980 年代に始めて提案され，以降様々な匿名通信システムが生まれた．中でも現在最も有名かつ普及しているのは，第二世代のオニオンルーティングの実装にあたる Tor[4] である．これは TCP プロトコル上の匿名通信を可能にし，同プロトコルを用いる各種アプリケーションにプライバシー保護を施すための基盤技術として機能する．

一方で，Tor の匿名性を低下させる有効な攻撃手段も発見されつつある．中でも入り口ノードのトラフィックのみに着目する指紋攻撃 [12] は，実現のために必要な資源が少なく，現実的な脅威になりうるものとして注目されている．しかしながら，Tor はトラフィックに様々な加工を施した上で通信を行うため，この手法は必ずしも十分には機能していない．また，静的な指紋攻撃であれば，対策を施すのも容易である．本研究ではこの加工を打ち消すような逆加工によって，Tor の安全性が更に低下することを示

す．そして，トラフィック逆加工の持つ潜在的な脅威について議論し，Tor が解決すべき今後の課題について考察する．

2 匿名通信システム

2.1 匿名通信システムの歴史

Chaum が提唱した Mixnet[2] や Dining cryptographers problem[1] が匿名通信システムの始まりであるが，この時にはまだノード情報の管理方法やパスの生成方法などが考慮されておらず，実装するには難があった．そこで後の研究者がこれらの課題に取り組み，その結果，安全に匿名通信を実現するシステムが多く提案された．現在最も普及しているのは，第二世代のオニオンルーティングにあたる Tor[4] である．Tor は，P2P 技術を利用した SOCKS プロキシとして動作し，通信の際には Tor ネットワークから無作為に数ヶ所のノードを選び，それらを中継させながらメッセージを送信する．ここで，ノードを中継させる際にオニオンルーティングと呼ばれる多重暗号化を行うことで，個々のノードやその通信を盗聴した第三者からは通信の送受信者情報が特定できないようになっている．また，多重暗号化には共通鍵暗号方式を用い，動作の高速化を図っている．現在世界中で 2000 以上の Tor ノードが動作しており，この規模の大きさが，Tor の匿名性を大幅に高める要因となっている．

2.2 匿名通信システムの用途

今日の Web サイトはより良質なサービスを提供するためにサイト訪問者に対してプロファイリングを行っている場合があり．例えばアマゾンの様な大規模な BtoC サイトでは，商品ページの閲覧履歴や商品の購入履歴を元に，より適切な商品広告を提供しようとする．あるゲームを購入すれば，同一のプラットフォームで動作する別のゲームや，あるいは同ジャンルのゲームを，推薦商品として表示する．この様な動作は，確かに売り手にとっては売上の増加につながり，買い手にとっては購買の利便性を向上させるかもしれない．しかしながら同時に，サイト訪問者のプライバシーを脅かす動作でもある．また，プロファイリングのために収集した情報はかならずしも正しく扱われるとは限らず，悪意ある人間によって不当に利用される危険性すらある．Tor は，このようなプロファイリング活動に対する有効な回避手段となる．

社会的にデリケートな問題を扱う際にも，Tor は役に立つ [3]．例えば，危険な事件から生き延びた人々が，生き延びたという事実を公にすること無く Web 上の掲示板やチャットルームで情報共有したい場合や，特定の病気を持つものが，身元を明かすこと無くその病状について情報交換をしたい場合などがあてはまる．あるいは，ジャーナリストが密告者や反抗勢力に対して安全に接触したい時や，企業が競争的分析を行う時にも役立つかもしれない．

さらには，大使館が本国と通信する場合や，警察が隠密に証拠を収集しようとする場合にも用いられることがあり，Tor が我々の実生活に与えている影響は極めて大きい．

3 匿名通信システムに対する攻撃手法

3.1 結託攻撃 (Sybil Attack)

一方で，Tor の匿名性を脅かす攻撃手法も多数提案されている．

オニオンルーティングシステムに対する攻撃手法として最も単純なものは，ノードを多数占拠することによる結託攻撃（Sybil Attack）[5] である．Tor は，通信に用いられた全ての中継ノードが結託することで送受信者情報を特定可能であるため，ノードを占拠すればするほど，特定可能な機会は増加する．例えば，N のノードで構成された Tor ネットワークがあった場合，その中の m のノードを占拠すれば，$(m/N)^2$ の通信を支配可能になる．

3.2 先行点攻撃（Predecessor Attack）

ノードを多数占拠することで可能な攻撃は，他に先行点攻撃（Predecessor Attack）[14] がある．ここで，あるノードが中継者としても送信者としても動作可能な状態にあると仮定する．この時，外部のノードはそのノードが今どちらで動作しているかを知ることができない．しかし，Tor は送信者として動作している場合，匿名性の確保のために定期的に経路情報を更新する．従って，中継者として動作している時と比べてより多くのノードと接続することになる．この特性に基づくと，ある複数のノードが他と比べてある特定のノードと接続する機会が多いとき，その事実はそのノードが送信者として動作している根拠とみなせるかもしれない．この様な統計的な情報に基づいて送信者匿名性を暴く手法が，先行点攻撃である．先行点攻撃も結託攻撃と同様に，ネットワーク中のノードを占拠すればするほど，攻撃の実現性は増加する．

3.3 タイミング攻撃 (Timing Attack)

しかしながら，これらの攻撃に共通する「ノードを多数占拠する」という仮定は非常に多くの資源を必要とする．そこで，多数でなく，ネットワークの入り口と出口の二箇所を占拠しているという仮定を考えることがある．オニオンルーティングシステムに対する攻撃を考えるときは，こちらの仮定の方が主流である．入り口のノード T1 は，ある通信経路 I の始点であり，ある送信者 A からのメッセージを受け取っている．一方で，出口のノード T2 は，別のある通信経路 J の終点であり，ある受信者 B にメッセージを送っている．ここで，T1 と T2 で生じた通信に，時間差で強い相関性が観測できたとしよう．T1 と T2 の強い相関性は，T1 を始点とする経路 I と T2 を終点とする経路 J が実は同一の経路であることを示す根拠とみなせるかもしれない．相関性が強ければ強いほど，I=J である確からしさも高まるだろう．そして I=J であれば，A は B にメッセージを送っていた事が分かる．この様にして，通信の送受信者情報を特定するものは，タイミング攻撃 (Timing Attack)[7, 8] と呼ばれている．

3.4 反射攻撃 (Replay Attack)

I と J の同一性を判定する手段は，タイミング攻撃だけではない．例えば，T1 が中継したパケットと全く同一のパケットを不正に複製して，再度中継したとしよう．TCP パケットにはシークエンス番号が割り振られているので，前回と同一の番号を持つパケットを送った場合，それは不正なパケットと見なされてエラー処理がなされる．すなわち，もし A が B にメッセージを送っていたのであれば,T1 がパケットを複製することで T2 は受信者 B からエラーの報告を受ける．この特性を生かすことで，I と J の同一性を判定する手法が反射攻撃 (Replay Attack)[10] である．

3.5 指紋攻撃 (Fingerprinting Attack)

ここまで紹介した攻撃手法は，確かに強力ではあるもののその実現可能性は低いとみなされ，現実的な脅威として注目されることは少なかった．より実現性を高めるために，入り口ノードのみを占拠した場合の攻撃が研究された．中でも指紋攻撃 (Fingerprinting Attack)[12] は，Tor に対する有効な攻撃手法である．

例えば Tor を用いて，あるウェブサイトにアクセスする場合を考える．一般的なウェブページは，画像ファイルやスクリプトファイルなど，多くの依存ファイルで構成されている．従って，ブラウザが web サイトにアクセスする際は，それらのファイルに対してもリクエストを行う．ウェブページ自体のサイズやそれに依存するファイルの総数，及び個々の依存ファイルのサイズはウェブサイト毎に異なるため，ウェブサイトにアクセスした際に生じる通信の流れも，サイト毎に異なったものになる．この通信の流れの中に生じるサイト独自の特徴（これを指紋と呼ぶ）を上手くとらえて，送信者がどこのウェブサイトにアクセスしているかを特定しようというのが，指紋攻撃の基本的な方針である．

指紋攻撃が他の攻撃と大きく異なる点は，占拠すべきノードが一ヶ所で良いという点である．必要とする仮定が他の手法と比べて非常に弱く，攻撃の実現性は高い．

また，指紋攻撃のための具体的なトラフィック分析技術はいくつか提案されているが，機械学習を併用した手法が特に成功している [9, 6].

4 本研究が想定する Tor への攻撃形態

まず，あるインターネットユーザ A の存在を仮定する．A は自分がどの様なウェブサイトを訪問しているかを第三者から秘匿するため，匿名通信システム Tor を用いる．一方で，この A に対して，匿名性を暴こうとする攻撃者 B の存在を仮定する．B は A の入り口 Tor ノード M を占拠しており，そこを流れるあらゆる通信を観測可能とする．前章で説明した様に入り口ノードのみを占拠するという仮定は現実的であり，考察する価値が十分にある．また，B は同時にクライアント B' を持ち，M を入り口ノードとして Tor 経由の Web サイトブラウジングも可能とする．

B が行うのは，機械学習を併用した指紋攻撃である．まず，B は事前に B' を用いて任意のサイト i にアクセスし，M で観測されるトラフィック情報 T_i を取得する．そして，T_i から指紋 F_i を生成し，F_i の集合を訓練データとして分類器 C に学習させておく．C は，ある指紋を入力とし，それが属するウェブサイトを出力するような多クラス分類器である．F_i は多次元のベクトルであり，その生成には以下の四つの手法を用いる．

HTML: ウェブサイトを訪問する際，一番初めにリクエストするのは必ず HTML ファイルである．なぜならウェブページに含まれる依存ファイルの情報は全て，HTML ファイル内に記述されているためである．従って，ウェブサイトからのレスポンスの中で最も早く送られてくるのも必ず HTML ファイルであり，この部分の通信量を指紋として抽出する．

Flow: パケットの流れは，トラフィックを特徴づける重要な情報である．パケットの向きが変化するたびに直前の向き変更時からの通信量を記録しておき，最後にそれらを大きさ毎に 6 分類して指紋として抽出する．

Number: パケットの総数を，入力と出力に分けて計算し，指紋として抽出する．

Size: パケットの総通信量を，入力と出力に分けて計算し，指紋として抽出する．

最後に，B は M を観測し，A との通信で生じたトラフィックに C を適用して A が訪問したサイトを特定する．

5 指紋攻撃に対する防御手法

第 3 章で，入り口ノードを占拠することによる指紋攻撃が Tor に対して有効であることを説明した．しかし一方で，Tor はこの様なトラフィック分析に着目した攻撃を弱めるいくつかの特徴を持っている．更に，指紋攻撃に対する防御手法もいくつか提案されている．

5.1 固定パケットサイズ (fixed packet size)

大きなサイズのパケットを送信するとき，その通信が送受信者情報を特定するための鍵となってしまう場合がある．例えば，あるウェブサイト B が 1MB の画像ファイルを含んでおり，他のサイトでは最大でも高々1KB の画像ファイルしか扱っていないとする．この時，あるユーザ A の入り口ノードを占拠し，そこで 1MB の TCP パケットが観測できれば，その事実は A が B にアクセスしたことを示す有力な根拠になるだろう．Tor はこの様な指紋攻撃への対策として，パケットを固定サイズに分割して送信する機能を備えている．固定パケットサイズで通信することで，トラフィックに特別な特徴が生じることをある程度防ぐことができる．

5.2 過通信 (overcommunication)

Tor ノードが行う通信は，クライアントの TCP 接続の中継だけではない．例えば，Tor は匿名

性の確保のために定期的に経路を変更するが，その際には現在の経路を切断するための処理，新しい経路を検索するための処理，そして，新たな経路を確立するための処理が必要である．すなわち，入り口ノードを占拠して Tor のトラフィックを観察すると，その中には「TCP 接続を中継するための通信」と，「Tor のネットワークを制御するための通信」が混在していることになる．前者の直後に後者が重なると，両者の区別は非常につきにくく，あたかも後者が前者の一部であるかの様に見えることがある．ユーザがウェブサイトを訪問するときも，web アクセスに関するトラフィックの直後に web アクセスと無関係なトラフィックが付加されることがある．この様な Tor の特徴は，指紋攻撃を難しくする要因となる．過通信による防御は，見かけのオーバーヘッドが生じないことが特徴である．あくまでロードが完了した後にダミーの通信を追加するものであるため，過通信が生じても通常と同じ時間でロードが完了する．そのため，ユーザ側から見たときに利便性は一切損なわれていない．

5.3 ランダム HTTP パイプライン (rondomized http pipeline)

Tor の公式ブログ [11] では，指紋攻撃への対策としてランダムな HTTP パイプライン接続が提案された．HTTP パイプラインは，HTTP/1.1 で導入された機能である．これは web サイトの様な多数のファイルで構成されたコンテンツを読みこむ際に，逐次的にリクエストとレスポンスを繰り返すのではなく，リクエストとレスポンスの処理を並列して行うものである．具体的にはあるリクエスト 1 を発行し，レスポンス 1 を待つこと無く，リクエスト 2 を発行してレスポンス 2 を待機するといった動作となる．このようにして，ネットワーク帯域を有効利用することを試みる．ランダム HTTP パイプラインとは，この時のリクエストの発行順序をランダムにすることによって，トラフィックに一貫したパターンが生じるのを防ぎ，指紋攻撃から守ろうという手法である．ランダム HTTP パイプラインを行うためには接続先の Web サーバが HTTP/1.1 に対応していなければならず，従って有効範囲は限定的である．

5.4 カモフラージュ (camouflage)

カモフラージュは，Panchenko ら [9] によって提案された指紋攻撃への防御手法である．基本的な考え方はダミーパケットの混入であるが，人工的なパケットでなく，他のサイトへのアクセスで生じたパケットを混入させる．具体的には，ユーザ A があるサイト B にアクセスする際，バックグラウンドで同時に無作為に選んだサイト C にもアクセスする．こうすることで，ユーザ A の入り口ノードを占拠している攻撃者がいたとしても，観測されるトラフィックには B に関するものと C に関するものが混在し，特徴を正確に抽出することが困難になる．Panchenko はカモフラージュによって，通信のオーバーヘッドが 2 倍近くに増加するが，指紋攻撃を大幅に弱められることを報告している．

6 トラフィック逆加工

第 4 章で示した攻撃形態は，第 5 章で示したようなトラフィック加工技術のために，必ずしも最善の効果を得られていない．そこで本研究が提案するのが，この加工を打ち消すためのトラフィック逆加工 (Anti-Transform) である．

6.1 逆加工関数

トラフィック逆加工は，観測したトラフィック T を入力として，変換したトラフィック T' を出力するような関数の形で実装する．関数にはパラメータを設定し，例えば過通信に対応する逆加工関数は，$T' = A(T, scope, ratio)$ と表現され，以下に示す Algorithm1 に従って動作する．$fetch\_new\_unit(T)$ は，IP トラフィック T から一定個数の連続した IP パケット群 U を取得する関数で，$get\_ratio(U)$ は U の入出力比，$get\_position(U)$ は U の T 内における位置を返す．また，パラメータ $scope$ は過通信が発生し

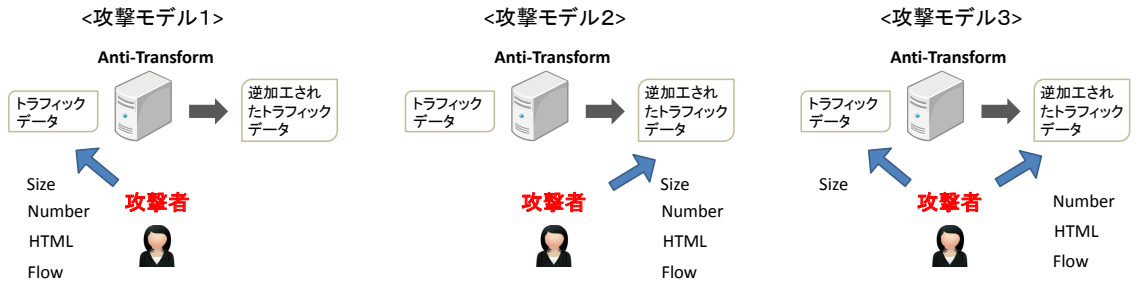

図 1: 逆加工を用いた攻撃モデルの例

Algorithm 1 Anti-Transform for overcommunication

$U \Leftarrow fetch\_new\_unit(T)$
while $U \neq null$ **do**
  $tmp\_ratio \Leftarrow get\_ratio(U)$
  $tmp\_position \Leftarrow get\_position(U)$
  if $tmp\_position$ is in $scope$ **then**
    if $tmp\_ratio \geq ratio$ **then**
      $T' \Leftarrow T' + U$
    end if
  else
    $T' \Leftarrow T' + U$
  end if
  $U \Leftarrow fetch\_new\_unit(T)$
end while

うる範囲を指定する値で，$ratio$ は過通信かどうかを判定する基準のパケット入出力比である．従って関数 A は，T の中で $scope$ の範囲内にありかつ $ratio$ の条件を満たすパケット群を除去し，結果を T’ として出力する．

6.2 適応性

逆加工の特徴の一つは，パラメータを用意することによって環境の変化に対する適応力を持つことである．一般的な攻撃手法は静的であり，実装が分かれば対策は容易といえる。一方で逆加工は動的であり，例えば既存の逆加工を防ぐ目的で Tor のトラフィック加工の実装が変更された場合，その新しい加工を適切に打ち消すように逆加工の実装も変化する．この様な動作は指紋攻撃の有効性を更に高め，Tor の匿名性に対する大きな脅威となる．

6.3 柔軟性

逆加工には他に，その利用形態を自由に選択できるという特徴がある．図 1 は攻撃モデルの外観であり，モデル 1 は逆加工を用いない標準的な指紋攻撃，モデル 2 は逆加工を施した上で行う指紋攻撃である．これらが最も単純なモデルだが，攻撃者は逆加工する前のデータと後のデータ双方にアクセス可能であるため，それらを組み合わせた攻撃も行うことができる．例えば，ある逆加工が一部の指紋攻撃には有効に作用するが一方でそうでない指紋攻撃も存在するとする．この時，有効に作用する指紋攻撃のみ逆加工されたデータで行い，残りは通常のデータで行えば良い．モデル 3 は，指紋攻撃の内 Number HTML Flow を逆加工されたデータ，残りを通常のデータで行う例である．
また，この図では単一の逆加工モジュールのみを仮定したが，指紋毎に複数用意することも可能であり，その場合更に柔軟な利用形態となる．

7 実験環境

第 4 章で示した攻撃形態に対応する，本研究での実装を説明する．インターネットユーザ A と攻撃者 B’ が使用するクライアントノードの情報と攻撃者 B が使用する入口ノードの情報を表 1 に記した．また，アプリケーションとして，

表 1: 実験環境

	クライアント	入り口ノード
OS	Linux(Ubuntu 11.10)	Linux(CentOS 6.2)
Tor	v0.2.1.30	v0.2.2.35

Web サイトの訪問には Firefox 13.0.1 を用い，IP パケットの観測には TcpDump を用いる．
分類器 C にはサポートベクトルマシン [13] を採用し, C への訓練データとして, B はサイトのランキング付けを行っている Alexa[1] の上位 50 サイトを B' から各 20 回ずつ訪問して計 1000 のトラフィックデータを取得する．

この後に, A も同様に Alexa の上位 50 サイトを 20 回ずつ訪問する．そしてその 1000 回の訪問の中で B が宛先を何回正確に特定できたかを計算し, この値を認識率（Recognition Rate）と定義し指紋攻撃の強さとみなす．逆加工関数には第 6 章の Argorithm 1 を用い, 各パラメータは認識率を報酬にした強化学習によって決定した．

8 実験結果

本研究の実験結果を表 2 に示す．1 行目の All は全ての指紋を用いた攻撃の結果であり，2 行目から 5 行目は，対応する指紋を単独で用いた攻撃の結果である．また，1 列目は逆加工を用いない攻撃モデル，2 列目は逆加工を用いた攻撃モデル，3 列目は逆加工を柔軟に活用した攻撃モデルの結果である．図 1 の攻撃モデル 1,2,3 が，それぞれ表 1 の Basic，Anti，Flexible Anti に対応する．

表 2: 実験結果．数値は認識率 (%)

設定	basic	Anti	Flexible Anti
All	74.48	75.71	**76.42**
HTML	4.38	4.38	4.38
Flow	65.91	66.42	66.42
Number	33.06	33.77	33.77
Size	65.81	64.59	65.81

[1] http://www.alexa.com/topsites/countries/JP

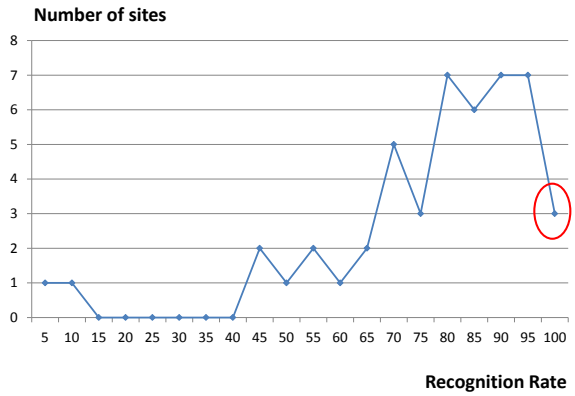

図 2: 認識率ごとの訪問先数

2 列目に注目すると，逆加工を用いることで All の認識率が増加し，指紋を個別に見ると Flow と Number で特に増加したことが分かる．しかしながら Size ではむしろ数値が低下したため，3 列目のモデルでは，Size には逆加工を適用せずに攻撃した．この Flexible Anti モデルで認識率は更に向上し，最終的に 76.42 の成果を得られた．最終的な成果を，訪問先で分類し認識率で整理した結果が図 2 である．半分以上の訪問先が高い認識率（80 %から 95 %）を記録し，更に完璧に匿名性が破られた（認識率が 100 %）訪問先が 3 つ存在した．この事実は本手法が Tor に対して極めて有効であることを示唆している．また，第 6 章で説明したようにこの手法は防御が難しい性質を持っており，既存の手法と比較して潜在的に持つ脅威は非常に大きい．

9 結論

本研究では，Tor がトラフィックに施す加工のために既存の指紋攻撃が十分に機能していないことに注目し，その加工を打ち消すようなトラフィック逆加工を提案した．そしてトラフィック逆加工が持つ特徴やその有効性について議論した．Tor は，インターネット上におけるプライバシーの保護を必ずしも保証しない．より強固な匿名通信を実現するためには，通信データを

Tor が安全に保護することを期待するのではなく，そもそも匿名性の高い通信データを扱うよう心がけねばならない．例えば，Web サーバー同士で結託して類似度の高いコンテンツ配信し個々の特徴が出ないようにする等は，指紋攻撃やトラフィック逆加工に対する有効な対策である．一方で Web サーバー同士で結託するという作業は非常にコストのかかるものでもあるので，このような問題を解決することも，Tor の今後の課題になっていくと考える．

参考文献

[1] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, Vol. 1, pp. 65–75, 1988.

[2] David Chaum, Communications Of The Acm, R. Rivest, and David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, Vol. 24, pp. 84–88, 1981.

[3] Roger Dingledine. Tor and circumvention: lessons learned., 2011.

[4] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *In Proceedings of the 13 th Usenix Security Symposium*, 2004.

[5] John Douceur and Judith S. Donath. The sybil attack. pp. 251–260, 2002.

[6] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naive-bayes classifier. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, CCSW '09, pp. 31–42, New York, NY, USA, 2009. ACM.

[7] Nicholas Hopper, Eugene Y. Vasserman, and Eric Chan-tin. How much anonymity does network latency leak. In CCS '07: Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007.

[8] Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew Wright. Timing attacks in low-latency mix systems (extended abstract). In *PROCEEDINGS OF THE 8TH INTERNATIONAL FINANCIAL CRYPTOGRAPHY CONFERENCE (FC 2004), KEY WEST, FL, USA, FEBRUARY 2004, VOLUME 3110 OF LECTURE NOTES IN COMPUTER SCIENCE*, pp. 251–265. Springer, 2004.

[9] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, WPES '11, pp. 103–114, New York, NY, USA, 2011. ACM.

[10] Ryan Pries, Wei Yu, Xinwen Fu, and Wei Zhao. A new replay attack against anonymous communication networks. In *ICC'08*, pp. 1578–1582, 2008.

[11] Tor Project. Experimental defense for website traffic fingerprinting, 2011.

[12] Yi Shi and Kanta Matsuura. Fingerprinting attack on the tor anonymity system. In Sihan Qing, Chris J. Mitchell, and Guilin Wang, editors, *ICICS*, Vol. 5927 of *Lecture Notes in Computer Science*, pp. 425–438. Springer, 2009.

[13] V. Vapnik. *Statistical Learning Theory*. Wiley, Chichester, GB, 1998.

[14] Matthew K. Wright, Micah Adler, Brian Neil Levine, and Clay Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Trans. Inf. Syst. Secur.*, Vol. 7, No. 4, pp. 489–522, November 2004.