結託的盗聴者の存在下で関係匿名性を実現する高速匿名通信プロトコルの設計

吉仲 佑太郎[1,a)]　武政 淳二[1]　小泉 佑揮[1]　長谷川 亨[2]

概要：大規模監視がプライバシーに対する攻撃であるということは，インターネットにおける共通認識である．しかし近年，国内外の政府機関や企業が監視を強化し，我々の日常的な通信の秘密を危機にさらしている．本稿では，こうした攻撃に対抗しインターネット上の日常的な通信における関係匿名性を保証することを目的として，超高速な匿名通信プロトコルを提案する．従来の匿名通信プロトコルとしては，各リレーがペイロード全体を復号するオニオンルーティングと，ペイロードの復号を一切行わない軽量匿名通信があるが，いずれもこの目的には適さない．オニオンルーティングは中継ノードにおける計算負荷が高く，コアネットワークへの実装が困難であり，軽量匿名通信は結託耐性を欠く．これに対して本稿で提案する，匿名通信プロトコルの新しいクラスであるキャベジルーティングでは，各中継ノードがペイロードの一部分のみを復号することで，結託耐性と高速性を合理的な仮定の上で両立させる．我々は，キャベジルーティングを具体化するプロトコルの構成とセキュリティ分析を行い，さらに我々の実装は計算機上でのオニオンルーティングからの高速化と，プログラマブルスイッチ上での Tbps 級の匿名通信を実証した．

High-Speed Anonymous Communication Protocol for Relationship Anonymity under Colluding Eavesdroppers

Yutaro Yoshinaka[1,a)]　Junji Takemasa[1]　Yuki Koizumi[1]　Toru Hasegawa[2]

Abstract: This paper proposes Cabbage Routing, a new class of anonymous communication, in which each relay decrypts only a portion of a packet payload, and adjacent relays collaboratively decrypt the entire payload. Our protocol achieves relationship anonymity under collusion while reducing computational overhead, demonstrating higher performance than Onion Routing on computers and Tbps-class anonymous communication on programmable switches.

1. はじめに

大規模監視はプライバシーに対する攻撃である [1].

そして，現にインターネットは大規模監視に直面している．米国では，国家機関とコンテンツプロバイダ (CP) やインターネットサービスプロバイダ (ISP) 等の複数の企業が結託し，大量かつ無差別にデータおよびメタデータを収集する，PRISM，Upstream プログラム [2,3] の暴露が，プライバシー研究の推進力となった．近年では，広告ネットワーク等を介して結託しユーザの意思に反して追跡を行う CP [4] や，顧客の通信に関する情報を第三者に提供する ISP [5] の存在が報告され，問題視されている．

本稿では，上記のような攻撃者を**結託的盗聴者**として定義する．第一に，彼らは結託的であり，すなわちネットワーク上の複数の地点を影響下に置き，各地点における監視情報を統合できる．例えば，悪意のある ISP と CP が結託するケースでは，ISP がユーザの身元を，CP が要求先コンテンツを把握する．これらの情報は観測されたパケット中の共通のビット列を通じて関連付けられ，結果として「特定のユーザが特定のコンテンツを要求した」事実が攻撃者の知るところとなる．

第二に，彼らは盗聴者，すなわちセミオネストであり，我々は受動的攻撃のみを想定する．これは本研究が，そもそも

[1] 大阪大学 大学院情報科学研究科
Graduate School of Information Science and Technology, The University of Osaka
[2] 島根大学 材料エネルギー学部
Faculty of Materials for Energy, Shimane University
[a)] y-yoshinaka@ist.osaka-u.ac.jp

実行に多大なコストを要する標的型監視 [6] ではなく，大量かつ無差別の情報を効率的に収集する大規模監視に注目し，その攻撃コストを引き上げることを目的とするためである．

大規模監視に対する代表的な対抗策としては．複数のリレーを経由して通信を行う匿名通信プロトコル，特に**オニオンルーティング** [7] が挙げられる．これは，各リレーがペイロード全体を再帰的に，タマネギの皮を剥がすように復号していくことで，攻撃者が各リレーの前後のパケットにおいて共通のビット列を見出すことを不可能にする，**ビットリンク不可能性**を実現する．この性質は結託耐性をもたらす一方で，各リレーにおいてパケットに対する複雑な暗号処理を要求し，性能劣化の原因となる．高性能化のために，オニオンルーティングをインターネットの AS 内部へ実装する方針 [8] が提案されているものの，暗号処理のオーバーヘッドが大きく，計算機上で高々 100 Gbps 程度の速度にとどまる．

これに対して，**軽量匿名通信** [9, 10] は同様に AS 内部に実装されるほか，リレーによるペイロードの復号を省略することで，性能劣化を最小限に抑える．これにより，計算機のみならず，プログラマブルスイッチ上で 最大 3.2 Tbps での動作を可能にする [11] が，結託に対して脆弱である．

本稿では，結託的盗聴者の存在下において，日常的な通信の妨げとならずに関係匿名性を保証することを目的とし，オニオンルーティングと軽量匿名通信を折衷する**キャベジルーティング** [12] を提案する．複数枚の葉が折り重なって中心部を覆うキャベツの構造に着想を得たキャベジルーティングでは，ペイロードの部分的な復号のみを行う複数のリレーが協力することで，ペイロード全体のビット列を変化させ結託耐性を実現する．ペイロードの一部分のみを処理する簡潔なプロトコルは，プログラマブルスイッチへのリレーの実装を初めて可能にし，Tbps 級の匿名通信を実現する．

我々は，まず匿名通信の関連研究を概観することでキャベジルーティングの動機を導き (2 章)，システムとセキュリティに関して厳密な問題設定を行う (3 章)．次に，キャベジルーティングの概念と，それを具体化するプロトコルの構成を示し (4 章)，その匿名性を形式的および確率論的に解析する (5 章)．さらに，このプロトコルに対して，高度に最適化されたソフト・ハードウェア的実装を与え (6 章)，それぞれについて性能評価を行う (7 章)．

2. 関連研究

本章では，脅威モデルと対応する技術，それに伴うプライバシーと性能のトレードオフに注目して，リレーに基づく匿名通信プロトコルを概観する (Table 1)．

2.1 1 ホップ匿名化—仮名化

単一のリンクの侵害のみを想定する，匿名性の最も単純な実現法は，ユーザに仮名を割り当て，単一のオネストなリレーにおいて実際の識別子と仮名を変換するものである．この変換は，単純な処理により実現され，ソフトまたはハードウェア的に回線速度で動作可能である．しかし，この方式は単一のリレーがオネストであることに依拠しており，そのリレーが侵害されると匿名性が完全に破られる．

2.2 軽量匿名通信—分離原則

軽量匿名通信は，経路上の単一のリンクまたはリレーのみを侵害する非結託的な攻撃者を想定し，インターネットの AS に実装される複数のリレーの連鎖を用いる．この方式の匿名性は，どのパーティも送信者と受信者の識別子を同時に観測できないという分離原則 [15] によって保証される．特に，能動的攻撃者の存在下における関係匿名性は，Bajic ら [10] によって議論された．非結託的な攻撃者を想定することの利点は，リレーによるペイロードに対する暗号処理が不要であるため，ソフトまたはハードウェアプラットフォーム上で高速な転送を実現できる点にある．しかし，結託に対して脆弱であり，大規模監視への対抗策としては十分でない．

2.3 オニオンルーティング—ビットリンク不可能性

オニオンルーティングは，経路上の複数のリンクおよびリレーを侵害する，結託的な攻撃者を想定し，ビットリンク不可能性を用いる．この性質は，各リレーにおいて入力パケットと出力パケットのビットパターンが暗号学的にリンク不可能であることを意味し，ペイロード全体の再帰的な復号によって実現される．しかし，この処理はリレーに多大な計算コストを課す，例えば，インターネットの AS に実装される最速のプロトコルである HORNET [8] であっても 100 Gbps 程度の速度にとどまり，インターネット基盤に求められる Tbps 級の転送の実現性には疑問が残る．

オニオンルーティングにおける能動的攻撃に対する防御は広く議論されてきたものの，実用的なプロトコルはしばしば意図的あるいは結果的に，これらの対策を組み込んでいない．例えば Tor [7] は，性能と引き換えに，出口ノード以外での完全性チェックを行わないため，能動的攻撃に対して脆弱である [16]．同様に HORNET も完全性チェックを省略しており，タグ付け攻撃に対して脆弱である [17]．能動的攻撃への対策には，MAC のパディングを伴う非展性暗号 [13] や可変長 Tweakable ブロック暗号 [18] などを必要とし，さらなる性能低下を伴うためである．

3. 問題設定

3.1 システムモデル

本プロトコルの実行には，通信の当事者である送信者と受信者に加えて，複数の**キャベジルータ (CR)** が関与する．送信者が通信を開始し，受信者が通信を受け入れる．CR は，送信者と受信者の間の通信を仲介する．

これらのパーティは全てインターネット上に存在し，特に CR は，匿名通信の提供に協力する自律システム (AS) に

			脅威モデル			性能	
プロトコルのクラス	例	技術	悪性リンク	悪性リレー	能動性	ソフトウェア	ハードウェア
1 ホップ匿名化	NAT, VPN	仮名化	単一	なし	あり	数 100 Gbps	10 Tbps
軽量匿名通信	dPHI [10]	分離原則	単一	単一	あり	100 Gbps	数 Tbps
キャベジルーティング	**本研究**	**部分的 B$\overline{\text{L}}$**	**複数**	**複数**	**なし**	**数 100 Gbps**	**Tbps**
オニオンルーティング	Tor [7], HORNET [8]	B$\overline{\text{L}}$	複数	複数	なし	100 Gbps	–
オニオンルーティング (形式的)	Camenisch ら [13]	ペイロード完全性	複数	複数	あり	–	–
ミックスネットワーク	Mixminion [14]	ミキシング	全て	一つ以外	あり	数 Mbps	–

Table 1: リレーベースの匿名通信プロトコルの比較

よって配置，運営される．各 AS は，管理下にある CR に関して，ソフトウェアまたはハードウェアの実装方式や，各パケットに対してどの程度の暗号処理を行うかを表す**復号長**を自ら選択し，コストと性能を調整することができる．

送信者は，公開ディレクトリから匿名で，CR のアドレスや復号長，AS 間のトポロジカルな関係やルーティング情報を取得できる．この情報の完全性は，公開鍵基盤 (PKI) が提供する証明書と公開鍵を用いてあまねく検証可能である．

3.2 脅威モデル

本プロトコルは，**結託的盗聴者**の存在を想定する．結託的であるとは，経路上に存在する複数のパーティを，ある上限数まで影響下に置くことを意味する．これには，CR の侵害，AS 全体の支配，受信者となることに加え，これらのパーティ間のリンク，すなわち IP ルータ，インターネットエクスチェンジ (IXP)，通信ケーブルの侵害を含む．盗聴者とは，受動的な攻撃のみを行う，セミオネストな攻撃者である．本研究が能動的な攻撃を想定しないのは，標的型監視ではなく大規模監視への対抗を動機とし，高性能と高ユーザビリティを通じて大きな匿名集合の形成 [19] を意図するためである．また本研究は，低遅延匿名通信で一般的であるように，ビット列以外の情報に基づく攻撃 (e,g., トラフィック分析，フィンガープリンティング) をスコープ外とする．

3.3 セキュリティゴール

本プロトコルは，**関係匿名性**の達成を目的とする．関係匿名性とは，送信者と受信者が互いに関連付けられないことを意味し [20]，送信者匿名性と受信者匿名性を包含する概念である [21]．この性質は，十分な匿名集合の確保のために必須の [8]，経路上の CR に対する**経路長秘匿性**を含む．

4. プロトコル

4.1 設計根拠

しばしばタマネギの皮を剥く操作に喩えられる，従来のオニオンルーティングでは，各リレーが毎回ペイロード全体に対して復号を適用する．しかし，この処理により実現される，各リレーの入出力パケット間のリンク不可能性は，想定する脅威によっては過剰となる場合がある．例えば，盗聴者が ISP と CP を影響下に置く場合には，各リレーの入出力ではなく，ISP と CP が観測するパケット間のリンク不可能性を確保すれば十分である．特に，プロトコルがインターネットの AS に実装される場合，経路上のリレー数が増大するため [8,22]，この過剰性はいっそう顕著となる．

これに対し，キャベツの構造から着想を得たキャベジルーティングでは，各リレーによる復号はペイロードの一部分に対してのみ適用され，連続する複数のリレーが協調的にペイロード全体を復号することで，その区間の両端におけるパケットのリンク不可能性を実現する．各リレーで実現されるこの性質を**部分的ビットリンク不可能性**と呼ぶ．例えば，各 CR が，ペイロードの $1/m$ 部分のみを復号する場合，m 個の連続する CR を通過する間にパケット全体のビットパターンが変化し，ビットリンク不可能性が成立する (l をデータ長として，$m = l/l_i$)．これを特に，**$1/m$-ビットリンク不可能性**（$1/m$-B$\overline{\text{L}}$) と呼ぶ．この方法により，各 CR での暗号処理による負荷を $1/m$ に抑えながら，経路選択が適切になされる限り，攻撃者の影響下にある複数地点間でのリンク不可能性が維持される．

4.2 経路選択アルゴリズム

CR は，あらかじめ自身の IP アドレス，ルーティング情報，利用する暗号スイートと復号長を，公開ディレクトリにアップロードしておく．この情報に基づいて，ある受信者 R と通信する意思を持つ送信者 S が，n 個の CR で構成される経路 $(N_1, ..., N_n)$ を選択する．

S は，まず自身の望むプライバシー強度に基づき，パラメータ α, β の値を決定する: α は経路上で攻撃者が影響下に置くことが想定されるリンクまたは受信者の最大数，β はそのような CR の最大数とする．次に，S はこれらの値と公開ディレクトリから取得した情報を用いて，経路を決定する．例えば，すべての CR が同じ復号長 m をサポートする特殊なケース (Figure 1) では，関係匿名性のためには，経路長 n が以下の条件を満たせば十分である:

$$n > (m-1)\alpha + m\beta - (m-1)$$

S は，効率的なアルゴリズムにより，上記の条件を満たす経路を発見できる．各 CR のサポートする復号長が異なるような一般の場合にも，S は，$O(n\alpha\beta)$ 時間・空間の動的

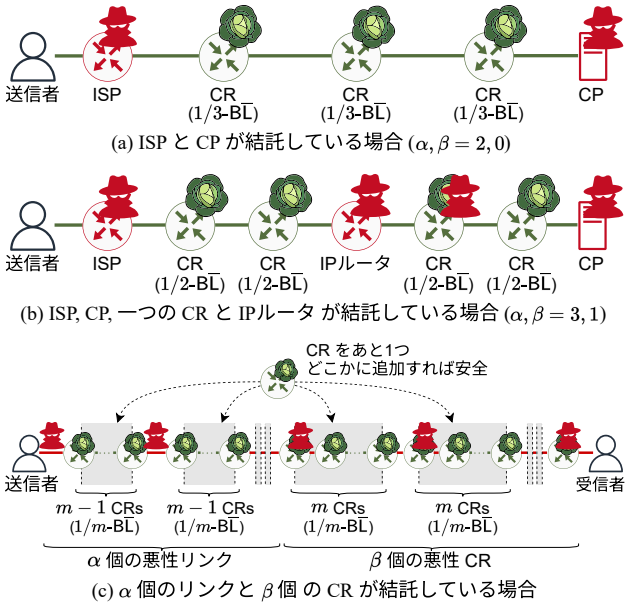


Figure 1: α と β に基づく安全な経路長の決定

計画法によって，与えられた経路が関係匿名性のために十分であるかどうかを判定できる．

4.3 経路設定フェーズ

キャベジルーティングにおける通信は，送信者 S が受信者 R および各 CR N_i との間で鍵を交換するとともに，N_i に対して後続のノード N_{i+1} との間で回路 ID を合意するよう要求する，**経路設定フェーズ**から開始される．この手続きは，Figure 2 に示される通り，前方秘匿性のためにホップごとの経路伸長 (テレスコーピング) により行い，鍵交換には証明可能安全性を持つ ntor [23] を用いる．経路伸長の際は，CR に対して経路長が漏洩してはならない．

はじめに，S は N_1 までの経路を確立するために，鍵ペア (x_1, g^{x_1}) を生成し，回路 ID cid_0 を選択して，cid_0 と g^{x_1} を持つ `Create` メッセージを N_1 に送信する．N_1 は鍵ペア (y_1, g^{y_1}) を生成し，y_1 と g^{x_1} に対応する共有鍵 $K_1 = (K_1^{\mathsf{PRP}}, K_1^{\mathsf{PRF}})$ と ntor の認証符号 t_1 を導出し．cid_0 と (g^{y_1}, t_1) を `Created` メッセージとして S へ返送する．

続いて，S は N_1 に，この経路の N_2 までの伸長を要求する．S は同様に N_2 宛に作成した `Create` パケットに対してパディングと K_1 を用いる暗号化によるカプセル化を施し，cid_0 とともに N_1 へ送信する．N_1 はこのカプセル化を解いた後，新しい回路 ID cid_1 を用いて，`Create` パケットを N_2 へ転送する．N_2 による `Created` パケットは N_1 で同様にカプセル化を受けてから S へ転送される．

この手続きを $n+1$ 回繰り返すことで，経路が R まで伸長され，各ノードが鍵と回路 ID と隣接ノードの情報を得る．

さらにこの手続きの中では，CR に対する経路長秘匿性のために，経路設定メッセージ (`Create` または `Created`) をデータパケットから識別不可能とし，CR がその個数を数えることを不可能にする．具体的には，N_{i+1} への経路伸長のための `Create` (resp. `Created`) メッセージに対して，N_i はペイロード全体の復号 (resp. 暗号化) を適用する．一方で，上流の CR $N_1, ..., N_{i-1}$ はこのメッセージに対して，通常のデータパットと同様に，部分的な復号 (resp. 暗号化) のみを施す．これにより，$N_1, ..., N_{i-1}$ はこのメッセージを通常のデータパケットから識別することができない．

Algorithm 1 データ転送フェーズにおける送信者 S

procedure onData$(data^{\rightarrow}, IP_{n+1}, IP_1, cid_0)$
  $(\_, \{l_i\}, \{K_i\}) \leftarrow \mathrm{Table}_0[IP_{n+1}, IP_1, cid_0]$
  $nonce_n \leftarrow \mathrm{GenNonce}() \times 2$
  $b_n^1 \parallel ... \parallel b_n^l \leftarrow \mathrm{Enc}(K_{n+1}^{\mathrm{PRF}}, data^{\rightarrow}, nonce_n)$
  for $i = n$ to 1 **do**
    $nonce_{i-1} \leftarrow \mathrm{PRP}^{-1}(K_i^{\mathrm{PRP}}, nonce_i)$
    $b_{i-1}^{l-l_i+1} \parallel ... \parallel b_{i-1}^{l} \leftarrow (b_i^{l-l_i+1} \parallel ... \parallel b_i^l) \oplus \mathrm{PRF}_{l_i}(K_i^{\mathrm{PRF}}, nonce_{i-1})$
    $b_{i-1}^{1} \parallel ... \parallel b_{i-1}^{l} \leftarrow b_{i-1}^{l-l_i+1} \parallel ... \parallel b_{i-1}^{l} \parallel b_i^1 \parallel ... \parallel b_i^{l-l_i}$
  end for
  $pkt_0^{\rightarrow} \leftarrow \left((IP_0, IP_1), \left(\rightarrow, cid_0, nonce_0, b_0^1 \parallel ... \parallel b_0^l\right)\right)$
  Transmit $pkt_0^{\rightarrow}$ to N_1.
end procedure
procedure onPktBwd$(pkt_0^{\leftarrow})$
  $\left((IP_0, IP_1), \left(\leftarrow, cid_0, nonce_0, b_0^1 \parallel ... \parallel b_0^l\right)\right) \leftarrow pkt_0^{\leftarrow}$
  $(IP_{n+1}, \{l_i\}, \{K_i\}) \leftarrow \mathrm{Table}_0[\_, IP_1, cid_0]$
  for $i = 1$ to n **do**
    $b_i^1 \parallel ... \parallel b_i^{l_i} \leftarrow (b_{i-1}^1 \parallel ... \parallel b_{i-1}^{l_i}) \oplus \mathrm{PRF}_{l_i}(K_i^{\mathrm{PRF}}, nonce_{i-1})$
    $b_i^1 \parallel ... \parallel b_i^l \leftarrow b_{i-1}^{l_i+1} \parallel ... \parallel b_{i-1}^l \parallel b_i^1 \parallel ... \parallel b_i^{l_i}$
    $nonce_i \leftarrow \mathrm{PRP}(K_i^{\mathrm{PRP}}, nonce_{i-1})$
  end for
  $data^{\leftarrow} \leftarrow \mathrm{Dec}(K_{n+1}^{\mathrm{PRF}}, b_n^l \parallel ... \parallel b_n^1, nonce_n)$
  if $data^{\leftarrow} = \perp$ **then**
    Abort.
  else
    Deliver $(data^{\leftarrow}, IP_{n+1}, IP_1, cid_0)$ to the upper layer.
  end if
end procedure

4.4 データ転送フェーズ

続く**データ転送フェーズ**で，送信者 S (=N_0) と受信者 R (=N_{n+1}) が，経路 $(N_1, ..., N_n)$ 上でデータを送受信する．

N_{i-1} と N_i の間を往路で転送されるパケット $pkt_{i-1}^{\rightarrow}$ を $pkt_{i-1}^{\rightarrow} = \left((IP_{i-1}, IP_i), \left(\rightarrow, cid_{i-1}, nonce_{i-1}, b_{i-1}^1 \parallel ... \parallel b_{i-1}^l\right)\right)$ と表す．このうち，前半のペアは IP ヘッダの送信元・宛先アドレス，後半のタプルはパケットの方向，回路 ID，ノンスと，各 64 バイトのブロックを l 個連接したペイロードである．

4.4.1 S におけるパケット送信

onData (Algorithm 1) は，送信者 S がデータ $data^{\rightarrow}$ を，(IP_1, cid_0) で特定される経路を用いて，受信者 N_{n-1} 宛に送信する処理である．送信者は，経路設定フェーズにの中で導出した情報をテーブルから取り出したあとで，鍵 K_{n+1}^{PRF} による認証暗号でエンド・ツー・エンドの暗号化を行う．この暗号文にはさらに，各 N_i と共有する鍵 K_i^{PRF} を用いて，繰り返し暗号化を適用する．具体的には，生成したノンスを鍵 K_i^{PRP} を用いる擬似ランダム置換 (PRP) によって順次更新しつつ，擬似ランダム関数 (PRF) を適用することで鍵ストリームを生成し，末尾 l_i 個のデータブロックを暗号化する．PRF_{l_i} は，出力長が l_i ブロックと等しい PRF である．最後に，この l_i ブロックがデータの先頭部となるようにブロック

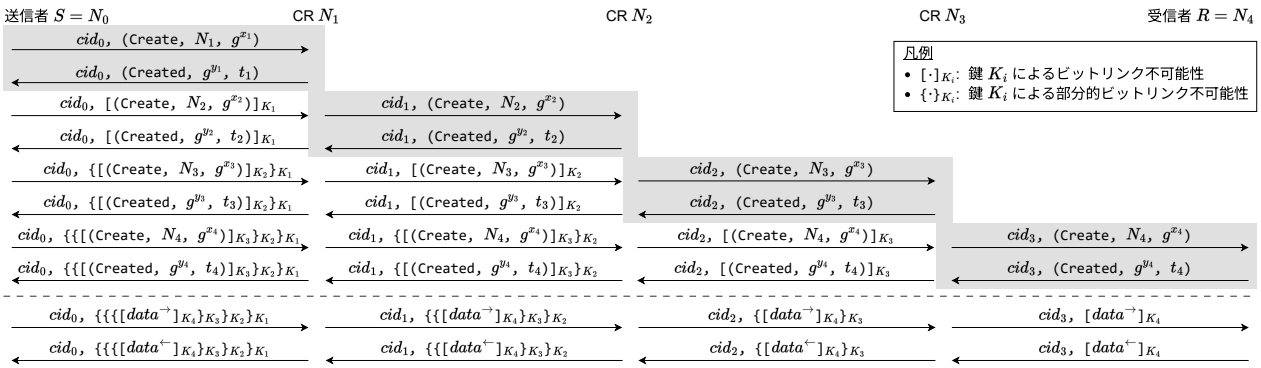


Figure 2: 経路設定フェーズとデータ転送フェーズにおけるメッセージ交換 ($n = 3$)

Algorithm 2 データ転送フェーズにおける CR N_i

procedure onPktFwd($pkt_{i-1}^{\rightarrow}$)
  $\left((IP_{i-1}, IP_i), \left(\rightarrow, cid_{i-1}, nonce_{i-1}, b_{i-1}^{1} \parallel ... \parallel b_{i-1}^{l}\right)\right) \leftarrow pkt_{i-1}^{\rightarrow}$
  $(\_, IP_{i+1}, \_, cid_i, (K_i^{\mathrm{PRP}}, K_i^{\mathrm{PRF}})) \leftarrow \mathrm{Table}_i[IP_{i-1}, cid_{i-1}]$
  $b_i^1 \parallel ... \parallel b_i^{l_i} \leftarrow (b_{i-1}^1 \parallel ... \parallel b_{i-1}^{l_i}) \oplus \mathrm{PRF}_{l_i}(K_i^{\mathrm{PRF}}, nonce_{i-1})$
  $nonce_i \leftarrow \mathrm{PRP}(K_i^{\mathrm{PRP}}, nonce_{i-1})$
  $pkt_i^{\rightarrow} \leftarrow \left((IP_i, IP_{i+1}), \left(\rightarrow, cid_i, nonce_i, b_{i-1}^{l_i+1} \parallel ... \parallel b_{i-1}^{l} \parallel b_i^1 \parallel ... \parallel b_i^{l_i}\right)\right)$
  Transmit $pkt_i^{\rightarrow}$ to N_{i+1}.
end procedure

procedure onPktBwd($pkt_i^{\leftarrow}$)
  $((IP_i, IP_{i+1}), (\leftarrow, cid_i, nonce_i, b_i^1 \parallel ... \parallel b_i^l)) \leftarrow pkt_i^{\leftarrow}$
  $(IP_{i-1}, \_, cid_{i-1}, \_, (K_i^{\mathrm{PRP}}, K_i^{\mathrm{PRF}})) \leftarrow \mathrm{Table}_i[IP_{i+1}, cid_i]$
  $nonce_{i-1} \leftarrow \mathrm{PRP}^{-1}(K_i^{\mathrm{PRP}}, nonce_i)$
  $b_{i-1}^{l-l_i+1} \parallel ... \parallel b_{i-1}^{l} \leftarrow (b_i^{l-l_i+1} \parallel ... \parallel b_i^l) \oplus \mathrm{PRF}_{l_i}(K_i^{\mathrm{PRF}}, nonce_{i-1})$
  $pkt_{i-1}^{\leftarrow} \leftarrow \left((IP_{i-1}, IP_i), \left(\leftarrow, cid_{i-1}, nonce_{i-1}, b_{i-1}^{l-l_i+1} \parallel ... \parallel b_{i-1}^{l} \parallel b_i^1 \parallel ... \parallel b_i^{l-l_i}\right)\right)$
  Transmit $pkt_{i-1}^{\leftarrow}$ to N_{i-1}.
end procedure

Algorithm 3 データ転送フェーズにおける受信者 R

procedure onPktFwd($pkt_n^{\rightarrow}$)
  $((IP_n, IP_{n+1}), (\rightarrow, cid_n, nonce_n, b_n^1 \parallel ... \parallel b_n^l)) \leftarrow pkt_n^{\rightarrow}$
  $K_{n+1} \leftarrow \mathrm{Table}_{n+1}[IP_n, cid_n]$
  $data^{\rightarrow} \leftarrow \mathrm{Dec}(K_{n+1}^{\mathrm{PRF}}, b_n^1 \parallel ... \parallel b_n^l, nonce_n)$
  if $data^{\rightarrow} = \perp$ **then**
    Abort.
  else
    Deliver $(data^{\rightarrow}, IP_n, cid_n)$ to the upper layer.
  end if
end procedure

procedure onData($data^{\leftarrow}, IP_n, cid_n$)
  $K_{n+1} \leftarrow \mathrm{Table}_{n+1}[IP_n, cid_n]$
  $nonce_n \leftarrow \mathrm{GenNonce}() \times 2 + 1$
  $b_n^1 \parallel ... \parallel b_n^l \leftarrow \mathrm{Enc}(K_{n+1}^{\mathrm{PRF}}, data^{\leftarrow}, nonce_n)$
  $pkt_n^{\leftarrow} \leftarrow ((IP_n, IP_{n+1}), (\leftarrow, cid_n, nonce_n, b_n^1 \parallel ... \parallel b_n^l))$
  Transmit $pkt_n^{\leftarrow}$ to N_n.
end procedure

を回転させ，以上の処理を各 CR に対して逆順に適用する．

4.4.2 N_i におけるパケット転送 (往路)

onPktFwd (Algorithm 2) は，CR N_i が往路におけるデータパケットの受信時に，ペイロードの一部を復号することで，部分的ビットリンク不可能性を実現する処理である．まず，受信したノンスに K_i^{PRF} を用いる PRF を適用して鍵ストリームを導出し，先頭 l_i 個のデータブロックを復号する．経路長秘匿性を保証しつつ，N_{i+1} 以降の CR が一貫して後続のブロックを処理することを可能にするために，復号されたブロックがデータの末尾となるようにブロックを回転させる．最後に，ビットリンク不可能性を破れを防ぐ目的で，ノンスに対して K_i^{PRP} を用いる PRP を適用する．

4.4.3 R におけるパケット受信・送信

onPktFwd (Algorithm 3) は，受信者 R が自分宛のデータパケットからデータを取り出し，受理する処理である．まず，鍵 K_{n+1}^{PRF} を用いてデータの復号と検証を行い，成功すれば平文を上位層に渡す．上位層から返送用のデータを受け取った際は onData に従い，データに対して，新たに生成したノンスと K_{n+1}^{PRF} を用いる認証暗号を適用する．

4.4.4 N_i におけるパケット転送 (復路)

onPktBwd (Algorithm 2) は，CR N_i が復路におけるデータパケットの受信時の処理である．K_i^{PRP} を用いる PRP によりノンスを更新した後，このノンスと K_i^{PRF} を使って末尾 l_i 個のデータブロックを暗号化し，暗号化されたブロックがデータの先頭部となるようにブロックを回転させる．

4.4.5 S におけるパケット受信

onPktBwd (Algorithm 1) は，送信者がデータパケットを復号してデータを取り出し，受理する処理である．まず，各 N_i に対して順に，鍵 K_i^{PRF} による部分的な復号とブロックの回転，K_i^{PRP} によるノンスの更新を行う．次に，鍵 K_{n+1}^{PRF} による復号と検証を行い，成功すれば平文を上位層に渡す．

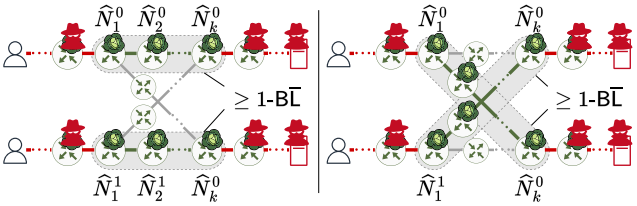
5. 安全性分析

5.1 形式的分析

本節では，キャベジルーティングのデータ転送フェーズにおいて，送信者 S と受信者 D がある十分条件下で関係匿名性を持つことを簡潔に示す．まず，連続する CR で構成される部分経路の両端におけるビットリンク不可能性を示す．さらに，その変形である**同時ビットリンク不可能性** [24] を経由して，関係匿名性の十分条件を導く．

5.1.1 ビットリンク不可能性 ($B\overline{L}$)

連続するオネストな CR で構成され，協力してペイロー



(a) 攻撃者が選択した2つの経路　(b) 送信者と受信者の入れ替え

Figure 3: 識別不可能なコンフィギュレーション

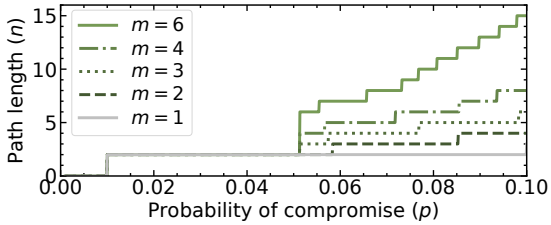


Figure 4: 99% の関係匿名性に十分な経路長

ド全体を 1 回以上復号する，盗聴されていない部分経路を，“良い” 部分経路と呼ぶこととし，$(\hat{N}_1, ..., \hat{N}_k)$ を長さ k の良い部分経路とする．この部分経路の両端における $B\overline{L}$ とは，攻撃者がプロトコルに従う任意のパケット列を 2 つ選択し，そのいずれかを復路において $\hat{N}_k$ から入力して $\hat{N}_k, ..., \hat{N}_1$ による暗号化を順に適用したあとで，その出力を再び攻撃者に提供するとき，攻撃者が入力されたパケット列を識別する際の優位性が無視可能であることをいう．この性質は，データ転送フェーズにおける CR のアルゴリズムの定義と，PRF と PRP の擬似ランダム性から導かれる．

5.1.2 同時ビットリンク不可能性 ($sB\overline{L}$)

$sB\overline{L}$ では，攻撃者が選択した 2 つのパケット列の両方に対して $\hat{N}_k, ..., \hat{N}_1$ による暗号化を適用し，両方の出力が攻撃者に提供され，攻撃者はどちらの出力がどちらの入力パケット列に対応するかを推測する．$sB\overline{L}$ は $B\overline{L}$ から導かれる．

5.1.3 関係匿名性 ($(SR)\overline{L}$)

$(SR)\overline{L}$ の定義 [21] では，攻撃者が受信者，CR とオネストな送信者から構成される 2 つの経路と共通のデータを選択する．2 つの経路は，確率 1/2 で送信者と受信者が入れ替えられたあとで，確立が行われ，データの転送が行われる．攻撃者がこれらの通信を盗聴し，入れ替えが行われたか否かを判定する際の優位性が無視可能であるとき，$(SR)\overline{L}$ が満たされているとする．キャベジルーティングにおいて，$(SR)\overline{L}$ が満たされるための十分条件は，以下の通りである．

1) 選択された 2 つの経路が，それぞれ，良い部分経路 $(\hat{N}_1^0, ..., \hat{N}_k^0)$ と $(\hat{N}_1^1, ..., \hat{N}_k^1)$ を含み，かつ
2) $\hat{N}_1^0$ と $\hat{N}_k^1$，$\hat{N}_1^1$ と $\hat{N}_k^0$ の間に，それぞれ，長さ k の良い部分経路が存在する．

これらの条件は，Figure 3 に表される，識別不可能な 2 つのコンフィギュレーションを生じる: (a) は攻撃者が選択した 2 つの経路がその通りに確立された場合で，条件 1) に相当し，(b) は 2 つの経路の送信者と受信者が入れ替えられて確立された場合で，条件 2) に相当する．これらのコンフィギュレーションの識別不可能性は，$sB\overline{L}$ から導かれる．

5.2 確率論的分析

経路長と匿名性の関係を調べるために，各リンクと CR が確率 p_α, p_β で独立に侵害を受けるとし，関係匿名性の破れる確率が 1% 以下となるような経路長 n の最小値を計算した．この確率は $O\left(\sum_i (l/l_i)\right)$ 時間と $O\left(\max\{l/l_i\}_i\right)$ 空間の BDD により計算可能で，またこのアルゴリズムは経路選択にも利用可能である．Figure 4 は，特に $l/l_i = m$ を定数とし，$p_\alpha = p_\beta = p$ としたときの，十分な経路長を表す．結果によれば，$p = 10\%$，$m = 2, 3, 4$ としたときの経路長はそれぞれ $4, 6, 8$ であり，HORNET が想定する経路長の範囲に収まる [8].

6. 実装

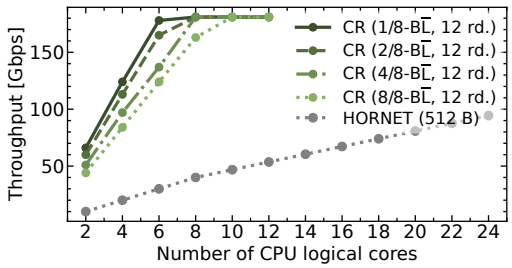
本章では，データ転送フェーズにおける CR の実装について，汎用計算機に基づくソフトウェアルータとプログラマブルスイッチ上での例を示す．PRF としては，暗号アクセラレータを持たないリソース制約下の送信者でも効率的に実行でき，かつ汎用計算機およびプログラマブルスイッチ上での効率的な実装が可能な ChaCha を共通して用いる．

6.1 ソフトウェア実装

PRF として ChaCha を，PRP として Crax-S-10 を採用し，AVX-512 拡張命令に対応する x86 ベース CPU 向けに CR を実装した．DPDK によりパケットはバッチ単位で受信され，まずテーブル参照に伴う DRAM アクセス遅延を隠蔽するためにプリフェッチ命令が実行される．このテーブルは，SipHash を用いるハッシュテーブルとして実装される．次に，同じバッチ内の各 16 パケットに対し，AVX-512 を用いた Crax-S-10 によるノンス更新を並列実行し，続いて各 4 パケットに対し，ChaCha による鍵ストリーム生成を並列に行う．OpenSSL をはじめとする既存の暗号ライブラリは，単一の鍵に対するバルクデータの処理に対して最適化されており，パケット間の並列化に対応しないため，これらの暗号は独自に実装・最適化した．最後に，データブロックの回転を行ったあとで，パケットを送出する．

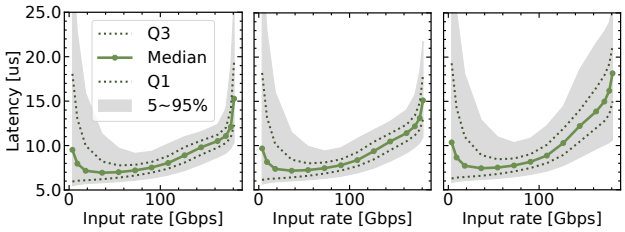
6.2 ハードウェア実装

PRF として ChaCha [25] を，PRP として 2EM [26] を採用し，Tofino 2 プログラマブルスイッチ ASIC 上に P4 言語で CR を実装した．受信されたパケットに対して，まず入力パイプラインにおいてテーブルから K_i^{PRP} を取得し，2EM の計算を行う．続いて K_i^{PRF} を取得したあと，再循環を伴いパケットをパイプラインに $4l_i$ 回繰り返し通過させることで，ブロックを回転させつつ ChaCha による鍵ストリームの生成を行う．なおプログラマブルスイッチ [27] は，特殊なアーキテクチャに起因する種々の制約を持つため，このような複雑



実装が未公開のため，HORNET の結果は論文 [8] から引用した．

(a) スループット (ChaCha12)



1/8-BL̄, 8 コア　4/8-BL̄, 8 コア　8/8-BL̄, 10 コア

(b) レイテンシの分布 (ChaCha12)

Figure 5: ソフトウェア CR の評価

な計算の実装法は自明ではない．これに対して我々は，2EM と ChaCha での変数の共有やパーサの計算能力の活用を含むテクニックを駆使して制約を回避し，実装を可能にした．

7. 性能評価

7.1 ソフトウェア実装の評価

7.1.1 スループット

方法. 復号長 l_i と処理に用いる CPU 論理コア数 をパラメータとして変化させ，スループットを計測した．評価環境としては，2 基の Intel Xeon Gold 6126 CPU と 2 枚の Intel E810 100 Gbps NIC を持つ計算機を使用した．外部の機器から最大 200 Gbps のテストトラフィックを入力し，CR における損失率が 10^{-3} 未満となる入力レートを二分探索した．

結果. Figure 5a は，ペイロード長を 512 バイト ($l = 8$ ブロック) とし，ChaCha12 を用いた場合のスループットを表す．結果は，ソフトウェア CR は CPU における計算がボトルネックであり，スループットはコア数に対して線形にスケールすることを示す．一般に，最大スループットは 6–10 論理コアで達成され，短い復号長は少ない論理コアで回線を飽和させることを可能にする．また，4 論理コアの CR が 20 論理コアの HORNET ルータに匹敵するスループットを達成していることは，評価環境の違いを考慮しても，キャベジルーティングの簡潔なプロトコルの利点を示している．

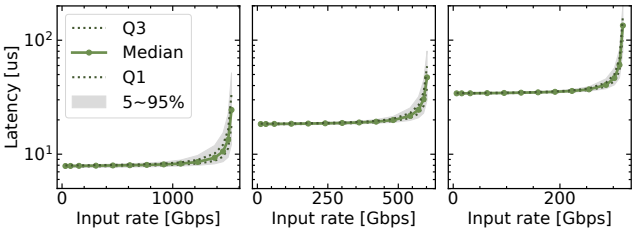
7.1.2 レイテンシ

方法. トラフィック生成器における送受信の時間差から得られるポート・ツー・ポートの遅延と，CR の CPU 内部で RDTSC 命令を実行することで得られる処理遅延を計測した．

結果. Figure 5b は，ペイロード長を 512 バイトとし ChaCha12 を用いる CR の ポート・ツー・ポート遅延の分布を示す．

l: #blocks (size)	l_i: 復号長 1	2	3	4	5	6
3 (192B)	1600, 1405	1107, 793	793, 552	–	–	–
4 (256B)	1600, 1466	1164, 823	823, 578	640, 439	–	–
5 (320B)	1600, 1493	1192, 850	850, 596	657, 455	540, 369	–
6 (384B)	1600, 1531	1218, 871	871, 602	677, 466	551, 379	466, 318

(a) スループット (ChaCha8, 12) [Gbps]



1/6-BL̄　3/6-BL̄　6/6-BL̄

(b) レイテンシの分布 (ChaCha12)

Figure 6: ハードウェア CR の評価

結果は，復号長の増加と，極端に速い・遅い入力レートが，計算時間とキューイング遅延の増加，バッチングの非効率化をもたらし，遅延を増加させることを示す．しかしながら，典型的な遅延は 5–20 マイクロ秒程度と低く抑えられている．CPU 内部における処理遅延は，復号長が $l_i = 1, 4, 8$ の場合に 381, 491, 570 サイクルであり，約 2000 サイクルを要する HORNET ルータと比較して低い．

7.2 ハードウェア実装の評価

7.2.1 スループット

方法. ペイロード長 l と復号長 l_i を変化させ，スループットを計測した．評価環境としては Tofino 2 ASIC を搭載する AS9516-32D プログラマブルスイッチを使用し，外部から最大 1.6 Tbps のテストトラフィックを入力した．

結果. Figure 6a に，ChaCha8 および ChaCha12 と，複数のペイロード長と復号長での最大スループットを示す．プログラマブルスイッチのアーキテクチャでは，パケット再循環の帯域がボトルネックとなるため，復号長による影響が性能に対して支配的である．ChaCha8 を利用し 1 ブロックのみを復号の対象とする場合，スループットは測定上限の 1.6 Tbps に達した．ChaCha12 を用いる場合や 2 ブロックを復号する場合でも，1 Tbps を超える速度における動作が可能であるが，復号処理の複雑化はスループットの低下の直接的な原因となる．この結果は，プログラマブルスイッチへの実装を可能にするプロトコルの簡潔さとともに，復号長の調整を可能にするキャベジルーティングの利点を裏付ける．

7.2.2 レイテンシ

方法. ハードウェア CR とトラフィック生成器を用いるテストベッドで，ポート・ツー・ポート遅延を計測した．

結果. Figure 6b は，ペイロード長を 384 バイト ($l = 6$ ブロック) とし，ChaCha12 を用いた場合のレイテンシの分

布を示す．スループットの 9 割程度までの入力レートにおいては l_i が支配的であり，レイテンシは 6–40 マイクロ秒の範囲に収まる．これを上回る高レートではキューの伸長に伴い，レイテンシが 100 マイクロ秒程度まで増加する．しかし，この遅延はインターネットの典型的なエンド・ツー・エンド遅延 (数十ミリ秒) と比較して十分に小さい．

8. おわりに

本稿では，結託的盗聴者を想定する脅威モデルの下で，新たな匿名通信プロトコルのクラスとしてキャベジルーティングを提案した．我々は，具体的なプロトコルの構成を示したうえで，関係匿名性の満たされる条件を形式的および確率論的に分析した．さらに，キャベジルーティングのリレーについて，高度に最適化された実装を示し性能評価を行った．我々のプロトコルは，汎用計算機上での実装において，オニオンルーティングに属する HORNET に対して優れた性能を示しただけでなく，プログラマブルスイッチへの実装を可能にし，結託耐性を持つつも Tbps 級の匿名通信を実現した．

本研究において提案したキャベジルーティングは受動的攻撃のみを想定しているが，タグ付け攻撃に代表される能動的な非匿名化攻撃は，より洗練された攻撃者の存在下において深刻な脅威となる．したがって，証明可能オニオンルーティングに関する先行研究 [13, 18, 28] を踏まえ，ペイロード全体の完全性を隣接 CR 間で協調的に検証する手法の検討が今後の研究課題として挙げられる．また，筆者は匿名性がネットワーク自体によって提供されるべき本質的な性質であることを確信しているが，近年のサイバー犯罪の深刻化を鑑みれば，悪意ある送信者による匿名性の悪用を防ぐことも重要な課題であると考える．このためには，特定のパーティに対して無条件の信頼と権威を置くことなく，一般の送信者の匿名性を保証しつつ不正行為の証明を可能にする，契約ベースの手法 [29] を発展させることが望ましい．

謝辞　本研究は，JST CRONOS JPMJCS24N3，JSPS 科研費 24KJ1629 の支援を受けたものである．

参考文献

[1] S. Farrell and H. Tschofenig, “Pervasive monitoring is an attack.” IETF RFC 7258, 2014.
[2] G. Greenwald and E. MacAskill, “NSA Prism program taps into user data of Apple, Google and others.” The Guardian, 2013.
[3] A. Clement, “NSA surveillance: Exploring the geographies of Internet interception,” *iConference*, 2014.
[4] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz, “The web never forgets: Persistent tracking mechanisms in the wild,” in *ACM CCS*, 2014.
[5] US Federal Trade Commission, “A look at what ISPs know about you: Examining the privacy practices of six major Internet service providers,” 2021.
[6] R. Barnes, B. Schneier, C. Jennings, T. Hardie, B. Trammell, C. Huitema, and D. Borkmann, “Confidentiality in the face of pervasive surveillance: A threat model and problem statement.” IETF RFC 7624, 2015.
[7] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The second generation onion router,” in *USENIX Security*, 2004.
[8] C. Chen, D. E. Asoni, D. Barrera, G. Danezis, and A. Perrig, “HORNET: High-speed onion routing at the network layer,” in *ACM CCS*, 2015.
[9] H.-C. Hsiao, T. H.-J. Kim, A. Perrig, A. Yamada, S. C. Nelson, M. Gruteser, and W. Meng, “LAP: Lightweight anonymity and privacy,” in *IEEE S&P*, 2012.
[10] A. Bajic and G. T. Becker, “dPHI: An improved high-speed network-layer anonymity protocol,” in *PoPETs*, 2020.
[11] Y. Yoshinaka, M. Kochiyama, Y. Koizumi, J. Takemasa, and T. Hasegawa, “A lightweight anonymity protocol at terabit speeds on programmable switches,” *Computer Networks*, vol. 253, no. 110721, 2024.
[12] Y. Yoshinaka, J. Takemasa, Y. Koizumi, and T. Hasegawa, “Minimal and fastest anonymous communication against colluding passive adversaries,” in *IEEE INFOCOM*, 2025.
[13] J. Camenisch and A. Lysyanskaya, “A formal treatment of onion routing,” in *Crypto*, 2005.
[14] G. Danezis, R. Dingledine, and N. Mathewson, “Mixminion: Design of a type III anonymous remailer protocol,” in *IEEE S&P*, 2003.
[15] P. Schmitt, J. Iyengar, C. Wood, and B. Raghavan, “The decoupling principle: A practical privacy framework,” in *ACM HotNets*, 2022.
[16] X. Fu, Z. Ling, J. Luo, W. Yu, W. Jia, and W. Zhao, “One cell is enough to break Tor’s anonymity,” in *Black Hat*, 2009.
[17] C. Kuhn, M. Beck, and T. Strufe, “Breaking and (partially) fixing provably secure onion routing,” in *IEEE S&P*, 2020.
[18] J. P. Degabriele and M. Stam, “Untagging Tor: A formal treatment of onion encryption,” in *Eurocrypt*, 2018.
[19] R. Dingledine and N. Mathewson, “Anonymity loves company: Usability and the network effect,” in *Workshop on the Economics of Information Security*, 2006.
[20] A. Pfitzmann and M. Hansen, “A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management,” 2010.
[21] C. Kuhn, M. Beck, S. Schiffner, E. Jorswieck, and T. Strufe, “On privacy notions in anonymous communication,” *PoPETs*, vol. 2, 2019.
[22] V. Liu, S. Han, A. Krishnamurthy, and T. Anderson, “Tor instead of IP,” in *ACM HotNets*, 2011.
[23] I. Goldberg, D. Stebila, and B. Ustaoglu, “Anonymity and one-way authentication in key exchange protocols,” *Designs, Codes and Cryptography*, vol. 67, 2013.
[24] A. Kate, G. M. Zaverucha, and I. Goldberg, “Pairing-based onion routing with improved forward secrecy,” *ACM TISSEC*, vol. 13, no. 4, 2010.
[25] Y. Yoshinaka, J. Takemasa, Y. Koizumi, and T. Hasegawa, “On implementing ChaCha on a programmable switch,” in *Workshop on P4 in Europe*, 2022.
[26] L. Wang, H. Kim, P. Mittal, and J. Rexford, “Programmable in-network obfuscation of DNS traffic,” in *NDSS DNS Privacy Workshop*, 2021.
[27] P. Bosshart, G. Gibb, H.-S. Kim, G. Varghese, N. McKeown, M. Izzard, F. Mujica, and M. Horowitz, “Forwarding metamorphosis: Fast programmable match-action processing in hardware for SDN,” *ACM SIGCOMM CCR*, vol. 43, no. 4, 2013.
[28] M. Backes, I. Goldberg, A. Kate, and E. Mohammadi, “Provably secure and practical onion routing,” in *IEEE CSF*, 2012.
[29] E. J. S. D. B. Jonathan and M. McCune, “A contractual anonymity system,” in *NDSS*, 2010.