一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
NS2019-108(2019-10)

[ポスター講演] MECキャッシュにおけるキャッシュポリューション攻撃の検知

芦原　大和†　　上山　憲昭†

† 福岡大学工学部電子情報工学科　〒814–0180 福岡市城南区七隈 8–19–1
E-mail: †tl161274@cis.fukuoka-u.ac.jp, ††kamiyama@fukuoka-u.ac.jp

あらまし　Web 閲覧や動画配信サービスを快適に行うため 5G セルラーネットワークの基地局にキャッシュサーバを設置することでバックホールのトラヒック負荷を低減する Mobile Edge Computing (MEC) が精力的に検討されている．しかし悪意を持ったユーザが意図的に低人気のコンテンツに多数の要求を行うことでキャッシュの効果を低下させるキャッシュポリューション攻撃 (CPA: cache pollution attack) の問題が指摘されている．CPA 発生時にはできるだけ迅速に攻撃ホストを特定し, キャッシュを保護することが重要である．そこで限られたメモリ量で高精度に攻撃ホストを特定するため, 本稿ではホスト ID とコンテンツ ID の組をキーとするブルームフィルタを用い CPA ホストを検出し防御する方式を提案する．そして計算機シミュレーションによる性能評価により，提案方式を用いることでキャッシュヒット率の低下を 30〜60%に抑えられることを示す．

キーワード　MEC，キャッシュポリューション攻撃，ブルームフィルタ

[Poster Presentation] Detection of Cache Pollution Attack in MEC Cache

Takakazu ASHIHARA† and Noriaki KAMIYAMA†

† Department of Electronic and Information Technology, Fukuoka University
8–19–1, Nanakuma, Jounan, Fukuoka 814–0180
E-mail: †tl161274@cis.fukuoka-u.ac.jp, ††kamiyama@fukuoka-u.ac.jp

Abstract　To reduce the load of backhaul networks and improve the quality of web browsing and video streaming services by placing cache servers at access points of 5G cellular networks, the mobile edge computing (MEC) has been investigated extensively. However, the cache pollution attack (CPA) which degrades the effect of caches by intentionally sending many requests to non-popular content items will be a problem in the MEC. Quickly detecting CPA hosts and protecting cache servers is important. Therefore, in this paper, we propose a method of accurately detecting the CPA hosts with using a limited amount of memory resources. The proposed method is based on a Bloom filter using the combination of host ID and content ID as keys. Through numerical evaluation, we show that the degradation of cache hit ratio is reduced to 30 - 60%.

Key words　MEC, cache pollution attack, Bloom filter

1. はじめに

5G セルラーネットワークにおいては動画コンテンツなどの大容量データが多く転送されることから，基地局間を接続するバックホールネットワークのトラヒック量が増大し，その設備投資コストの増大や品質の低下が懸念される．そこでバックホールネットワークのトラヒック量の増加を回避するため，基地局にキャッシュサーバを配置し，ユーザからの配信要求に対しキャッシュサーバからコンテンツを配信する Mobile Edge Computing (MEC) が注目され，精力的に研究が進められている [3] [6]．ユーザの配信要求に対し，要求コンテンツが MEC でキャッシュされていれば，オリジンサーバや CDN のキャッシュサーバから配信しないで MEC から直接ユーザに配信できるため，バックホールネットワークのトラヒック量の低減が期待される．しかしキャッシュサーバの容量は有限であるため存在する全てのコンテンツを MEC でキャッシュすることはできず，新たにコンテンツをキャッシュする際に容量が不足する場合にはキャッシュ置換アルゴリズムを用いて選択したキャッシュ済みコンテンツの一部を削除する．通常，コンテンツの人気には偏りがあるため，高人気のコンテンツを優先的にキャッシュに

残すことでキャッシュヒット率の向上とトラヒック量削減効果の向上が期待できる．そのため最後に要求されてからの経過時間が最大のコンテンツから削除する LRU (least recently used) がキャッシュ置換アルゴリズムとして広く用いられている．

ところで悪意を持ったユーザが意図的に低人気のコンテンツに多数の要求を行うことで，MEC のキャッシュの効果を低下させるキャッシュポリューション攻撃 (CPA: cache pollution attack) の問題が指摘されている [4] [8]．CPA には，Locality-disruption 型と False-locality 型の 2 種類の攻撃法が存在する．一般にキャッシュの性能はコンテンツの人気の偏りが強いほど大きくなるため，Locality-disruption 型の CPA では攻撃ホストは全てのコンテンツに対して一様に要求を発生させることで，コンテンツの人気の偏りを緩やかにすることでキャッシュの効果を低下させる．一方 False-locality 型の CPA では，攻撃ホストは低人気のコンテンツに対して多数の要求を発生させコンテンツの人気の順位を入れ替えることで，正常ユーザの配信要求に対するキャッシュヒット率を低下させる．

多数の基地局に対しキャッシュメモリを設置する MEC においてはキャッシュの容量は限られていることから，低人気のコンテンツをキャッシュ対象とさせる False-locality 型 CPA の脅威がより大きいと考えられる．そこで本稿では，False-locality 型 CPA を行うホストを迅速に検知することで CPA の被害を防ぐ方式を提案する．限られたメモリを用いて迅速に CPA ホストを検知することが重要なことから，提案方式はホスト ID とコンテンツ ID の組をキーとする Bloom Filter を用いて CPA ホストを検知する．そして計算機シミュレーションにより，提案方式の有効性を確認する．以下，2 節では関連研究について述べ，3 節では提案方式の詳細を述べる．そして 4 節で性能評価結果について述べ，最後に 5 節で全体をまとめる．

2. 関 連 研 究

情報指向ネットワーク (ICN: information-centric networking) を対象とした CPA の検知技術が数多く提案されている．例えば Yao らは，コンテンツを要求比率と要求の平均発生間隔に基づき，コンテンツを人気コンテンツと不人気コンテンツの 2 つにクラスタリングし，人気クラスタから不人気クラスタに移ったコンテンツ数の比率が閾値を超えた場合に CPA の発生を検出し，移ったコンテンツに対する Prefix を持つ配信要求 (Interest) に対してはコンテンツをキャッシュしないことを提案している [9]．また Guo らは攻撃者の Interest は特定のルータから生成されるため Interest の経路の多様性が低いことに着目し，要求が均一に発生する場合と比較した経路の多様性が閾値を下回る場合に検知する方式を提案している [2]．また Conti らは，ランダムにサンプルしたコンテンツセットに対し，各コンテンツの長期間における要求発生比率と，あるスナップショットにおける要求発生比率との差の総和が閾値を超えたときに CPA の発生を検知する方式を提案している [1]．また Xu らは，CPA は同一の Prefix を用いて異なる多数のコンテンツが要求される点に着目し，複数のハッシュ関数とビットマップを用いて異なり数を計測する Flajolet-Martin sketch (FM Sketch) を用いて，異なり数が閾値を超えたときに CPA を検知する方式を提案している [7]．さらに Park らは，CPA 発生中はキャッシュされているコンテンツのランダム性が増加する点に着目し，コンテンツ名から得られるハッシュ値でキャッシュコンテンツを 2 値行列にマッピングし，高人気コンテンツを過去の 2 時点の行列との XOR をとって消去した行列のランクの統計的な量を閾値と比較することで CPA の発生を検知することを提案している [5]．しかしこれらの研究は ICN のルータのキャッシュに対する CPA を対象としており，MEC のキャッシュに対する CPA の検知に用いることができない．

MEC のキャッシュに対する CPA を対象とした研究としては，Paul らの研究 [4] や Yang らの研究 [8] が見られる．Paul らは，HetNet の Femtocell に設置されたキャッシュを攻撃対象とする CPA の影響をシミュレーション評価している [4]．また Yang らは，CPA が Cooperative MEC の性能に与える影響を計算機シミュレーションにより分析し，攻撃者が全ノードをカバーする最小数のキャッシュに攻撃を行った場合の効果を明らかにしている [8]．しかしこれらの研究は CPA の影響の計算機シミュレーションによる評価結果を示すのに留まっており，CPA の検知方式関する提案は見られない．

3. 提 案 方 式

正常なユーザは短い時間内に同一のコンテンツを複数回，視聴する可能性は低い．一方，CPA の攻撃ホストは短い時間内に同一コンテンツに対し複数回の視聴要求を発生させる．そのため各配信要求に対し，ホスト ID とコンテンツ ID の組をテーブルで管理し，同一ホストからの同一コンテンツに対する配信要求を検知した際に，CPA として検知すればよい．しかし全ての配信要求に対しホスト ID とコンテンツ ID の組をエントリとしてテーブルに収容し，さらにユーザの各視聴要求に対し管理テーブルの全てのエントリを検査する必要がある．そのためユーザ数とコンテンツ数の増加に伴い，管理テーブルに必要となるメモリ量とメモリアクセス回数が増加する．そこでメモリの消費量とメモリアクセス回数を抑えながら効率的にキーの存在判定が可能であるブルームフィルタを採用する．

ブルームフィルタは空間効率のよい確率的データ構造であり，k 個のハッシュ関数を用いて Key に対する k 個のハッシュ値を生成する．初期状態では全ビットがゼロにセットされた Bitmap に対し，生成されたハッシュ値に対応する Bitmap 上の k 個の Bit の値を確認し，すべて 1 であれば過去に出現した Key と判定し，1 つでも 0 があれば過去に 1 度も出現したことがない Key と判定する．そして Bitmap の生成された k 個のハッシュ値に対応する Bit を 1 にセットする．しかし異なる Key に対し同一のハッシュ値が生成され，新規のものが既出現と判断される偽陽性による誤検出の可能性がある．

本稿では，ホスト ID とコンテンツ ID の組をキーとして，ブルームフィルタを用いて同一ホスト・同一コンテンツ組の視聴要求の既出現性を判定することで，メモリサイズとメモリアクセス回数を抑えた CPA ホストの検知方式を提案する．ブルームフィルタの Bitmap のサイズ B を

$$B = -\frac{n \log p}{(\log 2)^2} \quad (1)$$

ハッシュ関数の個数 k を

$$k = \frac{m}{n} \log 2 \quad (2)$$

により設定する [4]．ただし p は，ブルームフィルタに n 個のキーが収容された時点における擬陽性による誤検知の発生確率である．

ブルームフィルタとは別に，検知した CPA ホストの ID を収容するテーブルを別途，用意する．CPA ホストリストは初期状態では空に初期化される．Bitmap 中の 1 の Bit が増加するにつれ，偽陽性による誤検出の確率は増加する．CPA ホストリストを用いるには，ブルームフィルタの bitmap が埋まるのを抑え誤検出を減らすためである．ユーザからの視聴要求に対し，まず CPA ホストリストに含まれるホストか否かの確認を行い，含まれないホストからの要求であれば，コンテンツとホスト ID でブルームフィルタを使い検索を行う．ブルームフィルタによって同ホストが同コンテンツを要求したと判断されなかった場合は，ブルームフィルタにコンテンツとホスト ID を記録し通常通りキャッシュを行う．ブルームフィルタによって同ホストが同コンテンツを要求したと判断された場合は CPA ホストリストに記録し，キャッシュ対象から外す．

4. 性能評価

M 個のコンテンツに対し，人気の高いものから順に $1, 2, \cdots, M$ の ID を付与する．キャッシュ置換方式は LRU を想定し，キャッシュサイズ 500，コンテンツ数 $M = 10,000$，ホスト数 $N = 10,000$ で 1,000 秒間シミュレーションを行う．$N_c = 10$ 個の CPA 攻撃ホストが存在し，各々はコンテンツの人気順位を把握していることを想定し，300 秒から 600 秒の期間，最も低人気の $C = 500$ 個，すなわち $(M - C \leq m \leq M)$ のコンテンツに対し CPA を行う．CPA ホストも正常ホストも各々，一定の平均値の指数分布に従う時間間隔で視聴要求を生成する．ただし正常ホストはパラメータ $\theta = 0.7$ の Zipf 分布に従いランダムに選択したコンテンツを要求する．一方 CPA ホストは，最も低人気の C 個のコンテンツに対しランダムに要求を行う Rand 型攻撃と，最も低人気の C 個のコンテンツを順番に要求する Smart 型攻撃を考える．提案方式は同一ホストが同一コンテンツを要求した際に CPA ホストとして検知するため，Smart 型攻撃は同一ホストが同一コンテンツを要求するまでの時間を最大化することで，提案方式に対し CPA の効果を最大化する攻撃方法である．コンテンツとホスト ID を掛け合わせた偽陽性の発生確率 $p = 0.01$，設計エントリ数を $n = 10^8$ で式 (1)(2) に従いブルームフィルタを設計した．CPA ホストの総要求発生レートは正常ユーザの総要求発生レートと等しく，100/秒とした．

4.1 CPA の時系列特性

図 1，2，3 に，CPA 対象コンテンツ数を $C = 100$，800，3,200 とした場合の，正常ホストの視聴要求に対するキャッシュヒット率の時系列グラフを示す．ただし提案方式を用いて CPA ホストを検知した場合と，用いない場合の結果を各々示す．$C = 100$(図 1) の場合，攻撃対象コンテンツ数が少ないため CPA の影響が小さく，CPA ホストの検知を行わない場合もキャッシュヒット率の低下度合いは小さい．しかし $C = 800$(図 2) や $C = 3,200$(図 3) の場合，CPA ホストを検知しないとキャッシュヒット率の大きな低下が見られる．提案方式を用いることで，$C = 800$ の場合は CPA を受けた直後は影響を受け正常ユーザのヒット率が低下するが，時間の経過に伴い CPA ホストが検知されキャッシュヒット率が回復する．しかし $C = 3,200$ の場合は CPA 対象コンテンツ数が多いため Smart 攻撃の場合は検知が十分に行われず攻撃時間内にキャッシュヒット率は回復しなかった．図 4 に $C = 3,200$ で Rand 型攻撃における結果を同様に示すが，提案方式を用いることで短時間に CPA ホストを検出でき，迅速にキャッシュヒット率が回復した．図 3 と図 4 を比較すると，Smart 型攻撃の方が提案方式の効果が低下している．また正常ホストであっても同一のホストが同一のコンテンツを要求する可能性があり，約 5.8%の正常ユーザが検知の対象となった．

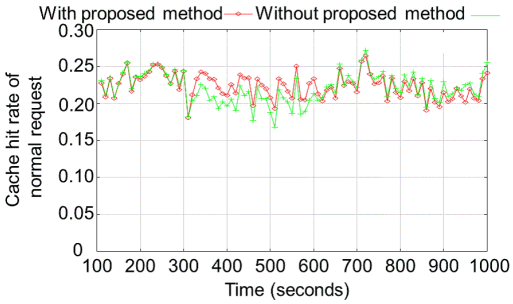

図 1 Time series of cache hit ratio of normal requests under smart CPA ($C = 100$)

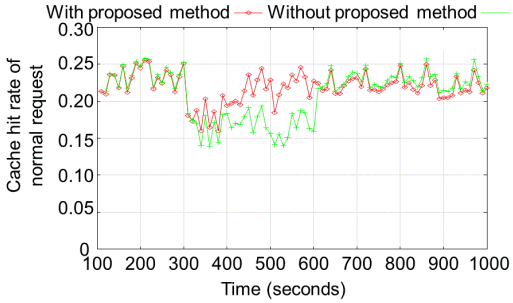

図 2 Time series of cache hit ratio of normal requests under smart CPA ($C = 800$)

4.2 CPA 対象コンテンツ数に対する特性

図 6 に，CPA 対象コンテンツ数 C を変化させたときの正常ユーザのキャッシュヒット率を示す．ただし normal は CPA が行われなかった場合，rand/bl は Rand 型 CPA が行われ提案方式を用いた場合，rand/cpa は Rand 型 CPA が行われ提案方式を用いなかった場合，*smart/bl* は Smart 型 CPA 攻撃が行われ提案方式を用いた場合，*smart/cpa* は Smart 型 CPA 攻撃が行われ提案方式を用いなかった場合の結果である．C の増加に伴い CPA の効果が増大し，CPA ホストの検知を行い場

合，キャッシュヒット率は低下する．一方，提案方式を用いることでキャッシュヒット率の低下が抑えられることが確認できる．しかし Smart 型攻撃に対しては，C の増加に伴い検知に要する時間が増加するため，提案方式を用いた場合もキャッシュヒット率が低下する．

また図 6 に C を変化させたときの，半数の CPA ホストの検知に要した時間 (av) と，全ての CPA ホストの検知に要した時間 (max) を各々を示す．C の増加に伴い検知に要する時間が増加し，特に Smart 型攻撃の場合，検知に要する時間が長い．

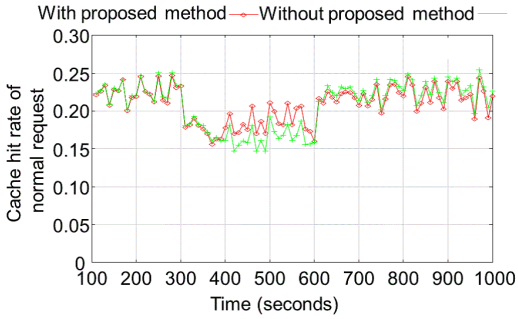

図 3　Time series of cache hit ratio of normal requests under smart CPA ($C = 3,200$)

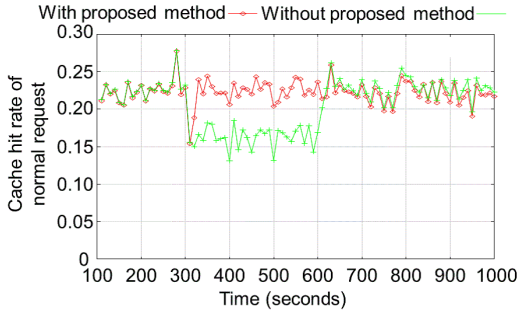

図 4　Time series of cache hit ratio of normal requests under random CPA ($C = 3,200$)

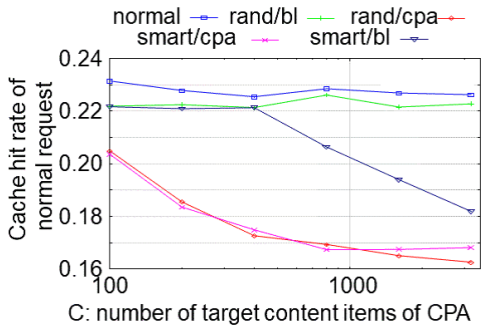

図 5　Cache hit ratio of normal requests against number of target content items of CPA

4.3　CPA 強度に対する特性

図 7 と図 8 に，CPA ホストの総要求レートの正常ユーザの総要求発生レートに対しする比率を変化させたときの，正常ユーザのキャッシュヒット率と検知に要した平均時間と最大時間を各々示す．提案方式を用いなかった場合，CPA ホストの総要求レートの増加に伴い正常ユーザのキャッシュヒット率は低下した．提案方式を用いた場合，CPA ホストの総要求レートの増加に伴い検出時間が減少するため正常ユーザが CPA から受ける影響は低下し，キャッシュヒット率は増加した．CPA ホストの総要求レートが低いほど CPA の検出に要する時間が増大するが，攻撃強度が弱いため正常ユーザのキャッシュヒット率の低下度合いは小さい．

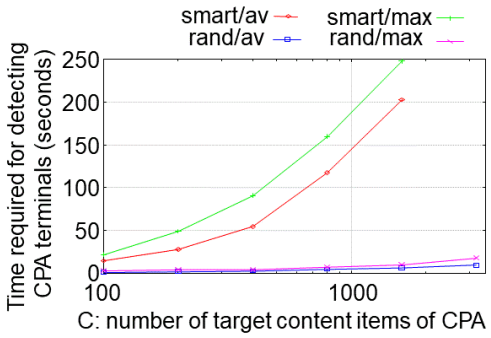

図 6　Time required for detecting CPA hosts against number of target content items of CPA

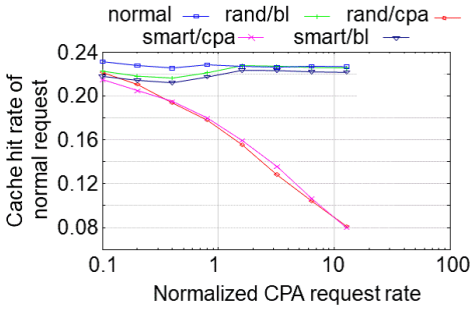

図 7　Cache hit ratio of normal requests against normalized CPA request rate

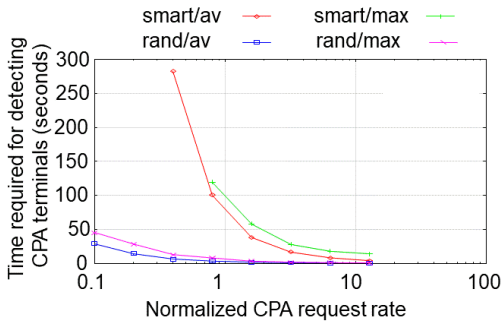

図 8　Time required for detecting CPA hosts against CPA request rate

4.4　攻撃端末数に対する特性

図 9 と図 10 に，CPA ホストの数 N_c を変化させたときの正常ユーザのキャッシュヒット率と検知に要した平均時間と最大時間を各々示す．提案方式を用いない場合，Smart 型攻撃では CPA ホストの増加に伴い，短期間に同じコンテンツが要求される頻度が増加しキャッシュの効果が増加するため正常ユーザのキャッシュヒット率は増加した．両攻撃方法とも CPA ホスト数の増加に伴い検出にかかる時間が増加するため正常ユーザ

のキャッシュヒット率が減少した．

4.5 正常ユーザの巻き添え率

図 11 に，正常ユーザのコンテンツ要求頻度分布の Zipf 分布のパラメータ θ を変化させるたときの，正常ユーザが提案方式によって CPA ホストとして誤検知された割合を示す．θ の増加に伴いコンテンツの人気度の偏りが大きくなるため，同一のホストが同一のコンテンツを要求する確率が増加するため正常ユーザの巻き添え率は増加した．

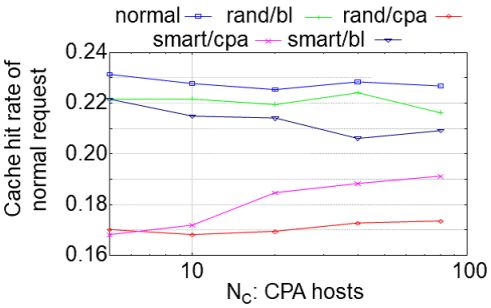

図 9 Cache hit ratio of normal requests against number of CPA hosts

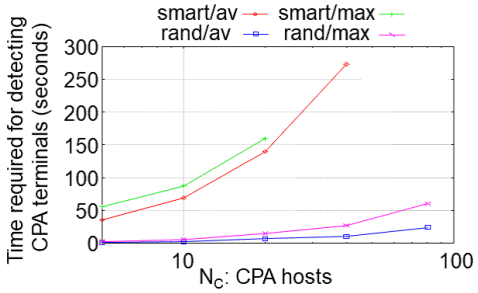

図 10 Time required for detecting CPA hosts against number of CPA hosts

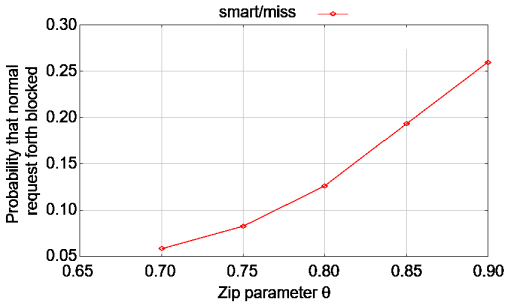

図 11 Probability that normal hosts are detected as CPA hosts

5. ま と め

本稿では False-locality 型 CPA を Bloom Filter を用いることで低コストで検知する方式を提案した．計算機シミュレーションにより，提案方式により正常ユーザのキャッシュヒット率の低下を 30〜60%程度，抑えることができることを確認した．今後は，検出に要する時間の短縮と正常ユーザの巻き添え率の低減のため，検出閾値を増加した場合やランダムにサンプルした要求に対してのみ検知を行う場合，さらに bitmap を一定間隔でランダムに選択し削除した場合の検討を行う予定である．

謝辞 本研究成果は JSPS 科研費 18K11283 の助成を受けたものである．ここに記して謝意を表す．

文 献

[1] M. Conti, P. Gasti, and M. Teoli, A lightweight mechanism for detection of cache pollution attacks in Named Data Networking, Elsevier Computer Networks, vol. 57, no. 16, pp. 3178-3191, Nov. 2013.

[2] H. Guo, X. Wang, K. Chang, and Y. Tian, Exploiting Path Diversity for Thwarting Pollution Attacks in Named Data Networking, IEEE Trans. Information Forensics and Security, vol. 11, no. 9, pp. 2077-2090, May 2016.

[3] J. Krolikowski, A. Giovanidis, and M. D. Renzo, Optimal Cache Leasing from a Mobile Network Operator to a Content Provider, IEEE INFOCOM 2018.

[4] S. Paul, A. Seetharam, A. Mukherjee, and M. K. Naskar, Investigating the Impact of Cache Pollution Attacks in Heterogeneous Cellular Networks, IEEE ICNP 2017.

[5] H. Park, I. Widjaja, and H. Lee, Detection of Cache Pollution Attacks Using Randomness Checks, IEEE ICC 2012.

[6] X. Wang, M. Chen, T. Taleb, et al., Cache in the Air: Exploiting Content Caching and Delivery Techniques in 5G Systems, IEEE Communications Magazine, vol. 52, no. 2, pp. 131-139, Feb. 2014.

[7] Z. Xu, B. Chen, N. Wang, Y. Zhang, and Z. Li, ELDA: Towards Efficient and Lightweight Detection of Cache Pollution Attacks in NDN, IEEE LCN 2015.

[8] C. Yang, H. Li, L. Wang, and D. Tang, Exploring the Behaviors and Threats of Pollution Attack in Cooperative MEC Caching, IEEE WCNC 2018.

[9] L. Yao, Z. Fan, J. Deng, X. Fan, and G. Wu, Detection and Defense of Cache Pollution Attacks Using Clustering in Named Data Networks, IEEE Trans. Dependable and Secure Computing, early access, Oct. 2018.