ICN における公開鍵暗号を用いたアクセス制御方式

Access Control Using Public Key Infrastructure in ICN

深川 悠馬　　上山 憲昭
Yuma Fukagawa　　Noriaki Kamiyama
福岡大学 工学部 電子情報工学科
Faculty of Engineering, Fukuoka University

1. はじめに

通信開始に先立つ名前解決を行わずにコンテンツの名称で要求パケットを転送し，ルータでコンテンツをキャッシュする ICN (information-centric networking) が，コンテンツを効率的に転送するネットワークとして注目されている．従来インターネットでは，有料や会員限定のコンテンツなど特定ユーザだけがコンテンツを消費できる配信サービスに対し，通信開始時に配信要求が必ずコンテンツ事業者 (Publisher) のサーバに届くため Publisher によるアクセス制御が可能である．しかし ICN では配信要求 (Interest) の転送経路上のルータで要求コンテンツがキャッシュされている場合，ルータから配信されるため Interest が Publisher に到達せず，Publisher によるアクセス制御が困難である．しかしルータでアクセス制御を行うには Publisher ごとのアクセス許可リストをルータで管理する必要があり，プライバシィ上，また処理負荷的に困難である．そこでアクセス権を有するすべてのユーザ (Consumer) に鍵を配布し，アクセス権を持たない Consumer はコンテンツを取得しても閲覧を不可能とする方式が提案されている [1]．本方式はコンテンツ単位ではなく，該当 Publisher のコンテンツ全体を対象にアクセス制御を同じ鍵で行うため，アクセス権を喪失した Consumer が閲覧不可とするには鍵を定期的に変える必要がある．そのため鍵を変えるたびにアクセス権を有する全 Consumer への鍵の再配布と更新が必要なので，処理・トラヒックのオーバヘッドが大きい．そこで本稿では，コンテンツの配信要求に先立ち，Consumer から公開鍵暗号方式で暗号化された Consumer の ID を含む Interest を常に Publisher に到達させることで，Publisher にてアクセス制御を可能とする方式を提案する．

2. 提案方式の特徴

データパケットのヘッダにキャッシュ可否を示すフラグ (CPF: cache permission flag) を用意し，データパケットを受信したルータは CPF が 1 の場合にのみデータパケットをキャッシュする．Consumer はコンテンツ m の先頭チャンク $I_{m,1}$ の要求に先立ちアクセス承認要求のための Interest パケット $I_{m,0}$ を送信するが，本パケットに対するデータパケット $D_{m,0}$ の CPF を 0 にセットし，コンテンツのチャンク $D_{m,1}$, $D_{m,2}$, …のパケット送信時は CPF を 1 にセットする．その結果，アクセス要求の Interest は常に Publisher に到達するため，Publisher は要求 Consumer のアクセス権の有無を判断し，アクセス権を有する場合にのみコンテンツを配信可能である．

Publisher はコンテンツを DES 等の共通鍵暗号方式で暗号化するが，Publisher は定期的に共通鍵を変更し，アクセス要求に対する応答として最新の共通鍵を要求 Consumer に配信する．そのため共通鍵の再配布を行うことなく，アクセス権を喪失した Consumer の視聴を防ぐことができる．提案方式では先頭チャンクの配信時にのみアクセス制御のオーバヘッドが発生するが，定期的に共通鍵を変えてもアクセス権を保有する全 Consumer に対する鍵の再配布が不要であり，オーバヘッド処理やトラヒック量を抑えることができる．

3. 提案方式の動作シーケンス

1. 事前にコンテンツ m の Publisher P_m は公開鍵基盤 (PKI: public key infrastructure) の自身の公開鍵 F_m を公開し，共通鍵 E_m を設定
2. Consumer c は m の要求に先立ち，自身の ID c を F_m で暗号化した $F_m(c)$ と，PKI の自身の公開鍵 F_c を，アクセス情報 $D_{m,0}$ を要求する Interest $I_{m,0}$ のペイロードに挿入して $I_{m,0}$ を送信
3. $D_{m,0}$ は経路上のルータのキャッシュに存在しないので，$I_{m,0}$ は P_m に必ず到達
4. P_m は $I_{m,0}$ から $F_m(c)$ を取り出し，自身の秘密鍵で復号して c を取得
5. P_m は c が許可された Consumer なら，E_m を F_c で暗号化した $F_c(E_m)$ を $D_{m,0}$ のペイロードに挿入し，$CPF = 0$ にセットして $D_{m,0}$ を送信
6. $D_{m,0}$ の中継ルータは $CPF = 0$ なので $D_{m,0}$ をキャッシュしないで c に向けて $D_{m,0}$ を転送
7. $D_{m,0}$ を受信した c は，$D_{m,0}$ から $F_c(E_m)$ を取り出し，自身の秘密鍵で復号して E_m を取得
8. c は m のチャンクに対する Interest $I_{m,1}, I_{m,2}, \cdots$ を送信し，該当データをキャッシュしているルータもしくは P_m は $CPF = 1$ にセットして，m のチャンク $D_{m,1}$, $D_{m,2}$, …を送信
9. $D_{m,1}$, $D_{m,2}$, …を受信した c は E_m で受信データを復号

4. 性能評価

提案方式は，アクセス要求 Interest の Publisher への送信と，Publisher でのアクセス判定処理と，共通鍵の Consumer への配信処理がオーバヘッドとして必要なため，オーバヘッドの処理時間 T は $T = \tau + W + S_1 + S_2$ となる．ただし τ は Consumer と Publisher との間の RTT，W は Publisher での処理待ち時間，S_1 は $F_m(c)$ の復号処理時間，S_2 は E_m の暗号化処理時間である．ここでは $S_1 = 1.5$ (ms)，$S_2 = 2.5$ (ms) とする [2]．

図 1(a) に，Amazon EC2 の 8 つのデータセンタから Public DNS server リスト [3] に掲載されている公開 DNS サーバに対して traceroute を行い測定した RTT の累積分布を示す．測定 RTT の平均値 198ms，5%値 20ms，95%値 350ms を，τ のサンプルとして評価に用いる．また Publisher には要求がレート λ(個/秒) のポアソン分布に従い到着し，Publisher のサービス時間は平均が $1/\mu = S_1 + S_2 = 4.0$ (ms) の指数分布に従うとすると，W は $W = 1/(\mu - \lambda)$ で与えられる．

図 1(b) に RTT の 3 つの値に対し，T を λ に対してプロットする．提案方式は Publisher への要求レートが約 250 個/秒より小さい場合には数 10ms〜数 100ms 程度のオーバヘッドでアクセス制御が可能であるが，要求レートがそれより増加すると急激にオーバヘッド時間が大きくなり鍵の配布が困難になる．今後は既存方式 [1] の全加入者に周期的に鍵を配布する場合と提案方式とで，オーバヘッドトラヒック量を比較する予定である．

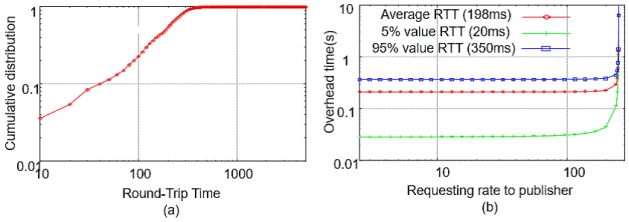

図 1: (a)RTT の累積分布 (b) 要求レートに対するオーバヘッド時間

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．

参考文献

[1] S. Misra, et al., AccConF: An access control framework for leveraging in-network cached data in the ICN-enabled wireless edge, IEEE Trans. Dependable and Secure Computing, 16 (1), Jan./Feb. 2019
[2] K. Xue, et al., A Secure, Efficient, and Accountable Edge-Based Access Control Framework for Information Centric Networks, ToN, June 2019
[3] Public DNS Server List, https://public-dns.info

Copyright © 2021 IEICE