1W-03

不正パケットの高速な検出を実現する簡易認証方式の提案と評価

鴨下 友馬 †1　鈴木 秀和 †1　内藤 克浩 †2　渡邊 晃 †1
†1 名城大学理工学部　†2 愛知工業大学情報科学部

1 はじめに

モバイルネットワークの普及に伴い，ネットワークセキュリティを脅かす DoS 攻撃 (Denial of Service Attack) が問題となっている．DoS 攻撃は，大量のデータを処理するサーバ類では特に脅威となる攻撃である．DoS 攻撃対策の一例として，共通鍵を事前に共有している場合は HMAC (Hash-based Message Authentication Code) を用いたパケット認証を利用することができる．しかし，この認証方式ではパケット長が長いと不正パケットの検出にかかる処理時間が長くなる．

そこで，共通鍵とシーケンス番号のみを用いた簡易認証方式を提案する．この方式では，共通鍵とシーケンス番号から生成した短いハッシュ値をパケット内に付加し，その値を最初に検証する．これにより，不正パケットのほとんどを高速に検出することが可能となる．

本稿では，実験による評価を行い，提案方式の有用性を示す．実験においては，移動透過性と通信接続性の両者を同時に実現する NTMobile (Network Traversal with Mobility) [1] を用いたので，これについても記述する．

2 既存のパケット検証処理

IPsec (IP security) の ESP (Encapsulating Security Payload) [2] トランスポートモードを用いて，パケット受信時の DoS 攻撃対策処理を説明する．図 1 にパケットフォーマットを示す．ESP header には 32bit のシーケンス番号が含まれている．認証コードは，パケットの認証範囲と共通鍵を用いて HMAC-MD5 により生成し，その結果を ICV (Integrity Check Value) に付与する．

パケット検証はリプレイ攻撃チェック，ICV(MAC) 認証の順に行い，不正パケットであると判定した場合にはその時点で破棄を決定し，以降の処理は行わない．ここで，リプレイ攻撃とは攻撃者が正規のパケットを盗聴し，それを再送する攻撃である [3]．リプレイ攻撃チェックはこれを阻止するための処理であり，リプレイ防御ウィンドウと呼ばれるビットマスクを用いて受信を許可するシーケンス番号の範囲を決定することで，リプレイ攻撃パケットを検出する．リプレイ攻撃では正規のパケットを攻撃に利用するため ICV(MAC) 認証では検出できず，

図 1　ESP のパケットフォーマット

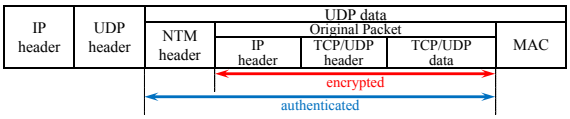

図 2　NTMobile のパケットフォーマット

リプレイ攻撃チェックは必須である．

ICV(MAC) 認証まで成功した場合は正規のパケットとみなし，リプレイ防御ウィンドウの更新を行う．最後に，パケットを復号して受信処理に入る．

3 NTMobile

NTMobile では，IPsec と同様にエンドツーエンドのセキュリティを実現することができる．NTMobile 端末は，通信開始時に DC (Direction Coordinator) からの経路生成指示によってトンネル経路を生成し，以後のすべての通信は仮想アドレスのパケットを実アドレスでカプセル化するという特徴がある．NTMobile には RS (Relay Server) という大量のパケットを中継する装置があり，不正パケットを可能な限り短時間で検出する機能が求められる．

図 2 に NTMobile のパケットフォーマットを示す．ここで，NTM header には NTMobile の通信に関わるデータが含まれており，32bit のシーケンス番号もここに含まれる．MAC には，HMAC-MD5 により生成した認証コードが付与される．パケット検証の方法は ESP と同様で，リプレイ攻撃チェック，MAC 認証の順に行う．

4 提案方式

提案方式はパケット検証の最初に行い，不正パケットを短時間で検出することを目的とする．したがって，パケット検証は簡易認証，リプレイ攻撃チェック，MAC 認証の順に行い，正規のパケットであればリプレイ防御ウィンドウの更新を行う．この方式は ESP，NTMobile 双方に適用できるが，パケット内に，簡易認証に係るフィールド（8bit）を追加する必要がある．

NTMobile の場合，送信側は，通信に用いる共通鍵とシーケンス番号を用いて 8bit のハッシュ値（以下，簡易ハッシュ値）を生成し，NTM header 内に格納する．ハッシュ関数は，演算時間の短い FNV-1(32bit) を用いることとする．受信側は，最初に簡易ハッシュ値を計算し，受信パケットに格納された値と比較する．これらの

Proposal and Evaluation of Simple Authentication Method that Detects Invalid Packets Fast

Yuma Kamoshita[†1], Hidekazu Suzuki[†1], Katsuhiro Naito[†2] and Akira Watanabe[†1]
[†1] Faculty of Science and Technology, Meijo University
[†2] Faculty of Information Science, Aichi Institute of Technology University

表 1　実験環境

OS	Ubuntu 14.04 32bit
CPU	仮想マシン（1Core 3.40GHz）
Memory	1GB

表 2　検証時間の実測値

$t_s[\mu s]$	$t_r[\mu s]$	$t_m[\mu s]$
0.536	0.414	3.835

値が不一致であれば不正パケットであると判定して破棄し，一致していればリプレイ攻撃チェックに進む．以後の処理は，既存のパケット検証処理と同様である．

5　実験と評価

5.1　検証時間の測定

パケット検証プログラムを試作し，表 1 の実験環境で処理時間の測定を行った．図 2 の authenticated の長さを 1,036Byte としてパケット検証処理を 100,000 回実行した際の，処理時間の平均値を表 2 に示す．ここで，t_s は簡易認証に要する時間，t_r はリプレイ攻撃チェックに要する時間，t_m は MAC 認証に要する時間である．

5.2　不正パケットの検証時間

不正パケットの検証時間を，実測値とシミュレーションから評価する．前提として，不正パケットの簡易ハッシュ値の分布は一様であると仮定する．このとき，簡易ハッシュ値の長さを l_h[bit] とすると，不正パケットにおいて簡易認証に成功する確率 P_s は，

$$P_s = \frac{1}{2^{l_h}} \tag{1}$$

となる．次に，シーケンス番号の長さを l_n[bit]，検証処理開始時点での最新の受信済みシーケンス番号を n_l，リプレイ防御ウィンドウのサイズを s_w，リプレイ防御ウィンドウ内の受信済みシーケンス番号の個数を $r\,(1 \le r \le s_w)$ とすると，不正パケットにおいてリプレイ攻撃チェックに成功する確率 P_r は，

$$P_r = \frac{\left\{\left(2^{l_n} - 1\right) - n_l\right\} + (s_w - r)}{2^{l_n}} \tag{2}$$

となる．したがって，不正パケットが簡易認証で破棄される確率 $\overline{P_s}$，リプレイ攻撃チェックで破棄される確率 $\overline{P_r}$，MAC 認証で破棄される確率 $\overline{P_m}$ は，

$$\overline{P_s} = 1 - P_s \tag{3}$$

$$\overline{P_r} = P_s\,(1 - P_r) \tag{4}$$

$$\overline{P_m} = P_s\,P_r \tag{5}$$

となる．以上より，不正パケットの検証時間の平均値 E は式 (6) のようになる．

$$E = t_s\overline{P_s} + (t_s + t_r)\,\overline{P_r} + (t_s + t_r + t_m)\,\overline{P_m} \tag{6}$$

以下，処理時間のシミュレーション結果を示す．$\overline{P_s}$ は定数であり，$\overline{P_r}, \overline{P_m}$ は式 (4)，式 (5) から P_r に依存し，$\overline{P_r}$ が小さいほど $\overline{P_m}$ は大きくなる．実験時の各種パラメータは簡易ハッシュ長 $l_h = 8$，シーケンス番号長 $l_n = 32$，リプレイ防御ウィンドウサイズ $s_w = \min(32, n_l)$ である．n_l, r は受信状況により変化するが，以下では $n_l = 1$, $r = 1$ とする．式 (6) で表 2 の実測値を用いると，

$$E = 0.553\,[\mu s] \tag{7}$$

表 3　正規のパケットの検証時間

$t_u[\mu s]$	簡易認証なし $[\mu s]$	簡易認証あり $[\mu s]$
0.561	4.810	5.346

表 4　簡易ハッシュ値の長さを変化させた場合の比較

l_h[bit]	8	16	32
$\overline{P_s}$	0.99609	0.99998	0.99999
$\overline{P_r}$	1.819×10^{-12}	7.105×10^{-15}	1.084×10^{-19}
$\overline{P_m}$	3.906×10^{-3}	1.526×10^{-5}	2.328×10^{-10}
$t_s[\mu s]$	0.536	0.675	0.823
$E[\mu s]$	0.553	0.675	0.823

となった．一方，式 (6) において $l_h = 0$, $t_s = 0$ とすれば，既存の不正パケット検証時間の平均値が求められる．よって，表 2 の実測値 t_r, t_m を用いると，

$$E|_{l_h=0,\,t_s=0} = 4.249\,[\mu s] \tag{8}$$

となる．この結果から，簡易認証により不正パケットの検証時間を約 1/8 に短縮でき，不正パケットによる DoS 攻撃耐性が大きく向上することが期待できる．

5.3　正規のパケットの検証時間

攻撃がない状態においては，パケット検証に常に簡易認証処理が加わるため負荷が増えることになる．そこで，この増加した負荷が全体の検証処理時間に与える影響を調査した．正規のパケットの検証処理を 100,000 回実行した際の，リプレイ防御ウィンドウ更新処理時間の平均値 t_u と，全体の検証処理時間の理論値を表 3 に示す．提案手法では t_s が加わるため全体の検証処理時間は約 11% 長くなるものの，この後の復号処理時間（MAC 認証の約 10 倍），さらには正規の受信処理時間を考慮すると影響は極めて小さいと言える．

5.4　簡易ハッシュ値の長さによる処理時間の比較

簡易ハッシュ長 l_h を変化させた場合の $\overline{P_s}, \overline{P_r}, \overline{P_m}$，および簡易認証を 100,000 回実行した際の処理時間の平均値 t_s，さらに不正パケットの検証時間の平均値 E を表 4 に示す．l_h が大きくなるほど，$\overline{P_s}$ は 1 に近付き，$\overline{P_r}$, $\overline{P_m}$ はほぼ 0 となる．このため，E は t_s の値に限りなく近付く．一方，l_h が長くなると t_s が増加し，その結果 E も増加する．このとき，t_s の増加量に対して，$\overline{P_s}$ の変化，すなわち簡易認証の効果は大差ない．したがって，簡易ハッシュ値の長さは 8bit で十分である．

6　まとめ

本稿では，共通鍵とシーケンス番号のみを用いた簡易認証方式を提案し，実験によりその有用性を示した．

参考文献

[1] 上醉尾一真ほか : IPv4/IPv6 混在環境で移動透過性を実現する NTMobile の実装と評価，情処学論，Vol. 54，No.10，pp.2288–2299 (2013).

[2] Kent, S.: IP Encapsulating Security Payload (ESP), RFC 4303, IETF (2005).

[3] Rescorla, E., et al: Guidelines for Writing RFC Text on Security Considerations, RFC 3552, IETF (2003).