汚染攻撃に耐性を持つXORネットワーク符号化の比較・評価

伊澤 和也† 宮地 充子† 面 和成†

†北陸先端科学技術大学院大学情報科学研究科
923-1292 石川県能美市旭台 1-1
{s0910201@jaist.ac.jp, miyaji@jaist.ac.jp, omote@jaist.ac.jp}

あらまし 2009年にYuらが提案したネットワーク符号化方式はXOR演算を使用した方式として初めて汚染攻撃への耐性を実現した．ここではノード間通信に鍵プール方式を採用し，完全性確保に使用されるMACの検証を転送ノードで行う．しかしながら，メッセージに処理を加える際，それに付加されたMACは未処理となる．

本稿では，まずセンサネットワークにおけるXORネットワーク符号化に関する2つの既存研究を比較・評価する．次に，Yuらの方式に対して，MACの数を減らすためにMACをアグリゲーションする改良を行い，その有効性を示した．

Comparison of XOR Network Coding against Pollution Attacks

Kazuya Izawa† Atsuko Miyaji† Kazumasa Omote†

†Information Science, Japan Advanced Institute of Science and Technology (JAIST)
1-1 Asahidai, Nomi, Kawasaki 923-1292, JAPAN
{s0910201@jaist.ac.jp, miyaji@jaist.ac.jp, omote@jaist.ac.jp}

Abstract In 2009, Yu et al. proposed a solution of XOR network coding, which is the first solution to pollution attacks. It uses probabilistic key pre-distribution and message authentication codes (MACs) for communications between nodes. In this way, forwarders verify the MACs to provide data integrity. But in the network coding process, the attached MACs leave unprocessed.

In this paper, we do comparative evaluation of two existing works on XOR network coding in wireless sensor networks. Furthermore, in order to reduce the number of MACs in the Yu's scheme, we performed improvement which aggregates MACs and show the validity.

1 はじめに

ネットワーク符号化は，送信ノードと受信ノードの間にある転送ノードでデータに符号処理を加えることで，ネットワークの効率的な利用（ネットワークの輻輳回避・スループットの向上）を実現するために考案されたデータ処理の一形態である．ネットワーク符号化では，単一送信ノードは複数の受信ノードに同じメッセージを送信することを想定する．ネットワーク符号化を行うノードは，受信したデータをXOR演算や線形変換を用いてデータに処理を加えて，データを1つにしたの後に次のノードにブロードキャストする．もちろん，ネットワーク符号化は最終的な受信ノードにおいて送信ノードが送付した個々のデータを全て復元可能である．

ネットワーク符号化に関する既存研究は数多くあるが，ここではXORネットワークコーデ

ィングのセキュリティを扱うことにする．Apavatjrut らの論文 [1] においては UHFs (Universal Hash Functions) ベースの MAC を使用したネットワーク符号化方式について，MAC のアグリゲーションについて検討している．一方，Yu らが提案した XOR ネットワーク符号化方式 [3] は MAC (Message Authentication Code) を用いて転送ノードでデータの完全性を検証することが出来る．

本稿では，センサネットワークにおける XOR ネットワーク符号化に関する 2 つの既存研究を比較評価する．さらに，MAC の偽造に耐性を持つ Yu らの方式において，同じ鍵が存在するときには対応する MAC 同士をアグリゲーションする単純な改良を行い，その有効性を示した．評価では，どの程度 MAC の削減が可能かの上限を導出し，Yu らの方式 [3] と同じパラメータで 2〜3 割削減できることを示した．

2 準備

2.1 汚染攻撃

汚染攻撃とは攻撃者によって送付データに対してデータの削除や挿入を伴うデータの改ざんを行う攻撃である．ネットワーク符号化においては複数のデータを送信して複数のノードにブロードキャストする性質上, 改ざんが行われたデータが複数の受信者にわたり, 汚染が広がる．ゆえにネットワーク符号化において，最終の受信者ではなく送信ノードが汚染されたメッセージを素早く見つけることが重要になる．

2.2 UHFs-based MAC

Brassard は，ワンタイムパッドを疑似乱数生成関数の出力に置き換えることによって，UHFs (Universal Hash Functions) から計算量的に安全な MAC アルゴリズムの構築について提案を行った [2]. UHFs とは鍵と呼ばれるパラメータに関連付けられた関数族であり, 全て異なる入力が衝突する確率が小さい．今回扱う UHFs の特定のクラスは，ϵ-almost XOR universal ハッシュ関数 (ϵ-AXU) である．ϵ-AXU は，$MAC_k(M_1 \oplus M_2) = MAC_k(M_1) \oplus MAC_k(M_2)$ を満たす準同型性を有する．ただし，M_1, M_2 は任意のメッセージであり, k は秘密鍵である．ϵ-AXU を用いた MAC では，$MAC_k(M) = f_k(M) \oplus r$ が成立する．ここで，f_k は UHFs であり, r はワンタイムパッドである．

3 既存研究

本章では，MAC のアグリゲーションを行う Apavatjrut らの方式と XOR ネットワーク符号化方式として初めて汚染攻撃への耐性を実現した Yu らの方式を概説する．

3.1 Apavatjrut らの方式 [1]

この方式では，UHFs ベースの MAC を用いた XOR ネットワーク符号化方式を提案している．論文では，具体的に 4 つのモードについて説明しているが，ここでは最も効率的な Xor-Authenticate-Forward (XAF) モードについて説明する．

3.1.1 パラメータの設定

この方式では, 単一の送信ノードと複数の受信ノードを想定する．ここでは，Alice, Bob, Eve の 3 ノードを想定しており，Alice と Bob は送信ノードかつ受信ノード，Eve は転送ノードである．Alice と Bob が Eve に各々のデータを送付した後, Eve が受け取ったデータを処理したものを Alice と Bob に送付するものとする．ϵ-AXU を用いた MAC 関数を以下に示す．

$$MAC_k(M) = f_k(M) \oplus r \tag{1}$$

ただし，k は全ノード共通の鍵である．

3.1.2 方式の詳細

XAF の模式図を図 1 に示す．Alice と Bob が各々 $d_1 = h_k(M_1), d_2 = h_k(M_2)$ を計算し，M_1, d_1 と M_2, d_2 を Eve に送信する． Eve が M_1', d_1' と M_2', d_2' を受信し, $M = M_1 \oplus M_2$ と

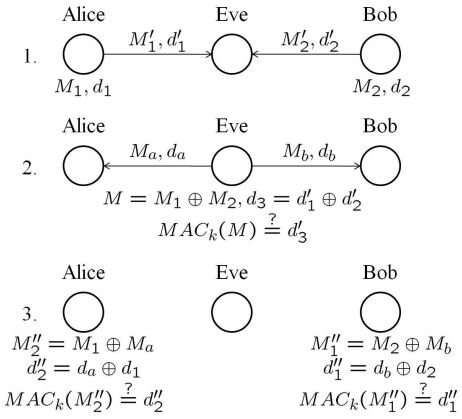

図 1: XAF(送信ノードと受信ノードが 2 つ)

$d_3 = d_1' \oplus d_2'$ を計算し, $h_k(M_1') \stackrel{?}{=} d_1'$ かつ $h_k(M_2') \stackrel{?}{=} d_2'$ ならば M, d_3 を Alice と Bob にブロードキャストする. データ確認は Alice と Bob が各々M_a, d_a と M_a, d_a を受信し, Alice は $M_2'' = M_1 \oplus M_a$ を計算し $h_k(M_2'') \stackrel{?}{=} d_2''$ が成立するならば, $d_2'' = d_a \oplus d_1$ を得る. Bob は $M_1'' = M_2 \oplus M_b$ を計算し $h_k(M_1'') \stackrel{?}{=} d_1''$ が成立するならば, $d_1'' = d_b \oplus d_2$ を得る.

3.2 Yu らの方式 [3]

ネットワーク符号化はスループットを最大化するために提案されたが, セキュリティ対策を実施しないそのままの利用では汚染攻撃に対して脆弱性を持っている. これまで汚染攻撃への耐性を有するネットワーク符号化方式は一般ネットワーク符号化のみの対応であり, XOR ネットワーク符号化には対応していなかった. XOR ネットワーク符号化は一般ネットワーク符号化と比較して計算量/通信量において有利であり, 既存研究では新たな XOR ネットワーク符号化方式を提案した. また, この方式では 単一の送信ノードと複数の受信ノードを想定する.

3.2.1 パラメータの設定

記号の表記を以下に記す.

- t: 各ノードが持つ鍵の数 (生成する $dMAC$ の数と同一)
- u: ハッシュに使用する符号語の数
- $k_{s,i}$: 送信ノードが持つ i 個目の鍵
- r_i: i 個目の $dMAC$ に使用されるハッシュチェーンのシード
- $r_{i,j}$: r_i のハッシュチェーンを計算した時の j 回目の要素.
- $Enc_{k_{s,i}}(), Dec_{k_{s,i}}()$: 共通鍵 $k_{s,i}$ を用いた暗号化と復号処理

t と u を公開パラメータとする. r_j からハッシュチェーンを計算可能な疑似ランダム順列関数 f ($[1,m] \to [1,m]$) とハッシュ関数 h ($Z_q^u \to Z_q, Z_q$ は符号語の範囲を制限) を導入し, 鍵プール K から t 個の $k_{s,1}, \ldots, k_{s,t}$ を抽出する.

3.2.2 dMAC の生成

送信ノードは, n 個のメッセージ $M_1, \ldots, M_n$ を複数の受信ノードに向けて送信し, 1 つのメッセージ M_i に対しては t 個の MAC を付加する. なお, ここで使用する MAC は復号可能なものであるため dMAC（decryptable MAC）と称する. 送信ノードは, 1 つのメッセージを送る際に以下を送信する.

$$M_i, id(k_{s,1}), dMAC_{i,1}, \ldots, id(k_{s,t}), dMAC_{i,t} \quad (2)$$

ただし, $dMAC_{i,j} = Enc_{k_{s,j}}(id(k_{s,j}), r_j, h_{i,j})$ であり, $id(k_{s,i})$ は $k_{s,i}$ のインデックスである.

次にハッシュ値の作成方法を説明する. まず, M_i を m 個に分割 ($M_i = m_{i,1}, m_{i,2}, \ldots, m_{i,m}$) する. 次に, f を用いて r_i を初期入力として次のように u 個の f-chain を作成する.

$$r_{j,v} = f(r_{j,v-1}) \quad (3)$$

ただし $r_{j,0} = r_j, v = 1, \ldots, u$ である. $r_{j,v}$ はハッシュに使用する符号語の場所を示す. ゆえにハッシュ値は以下のように計算される.

$$h_{i,j} = m_{i,r_{j,1}} \oplus \ldots \oplus m_{i,r_{j,u}} \quad (4)$$

図 2 は送信ノード S_i, S_j から $E = M_i \oplus M_j$ を受信した転送ノード CF が受信ノード R に対

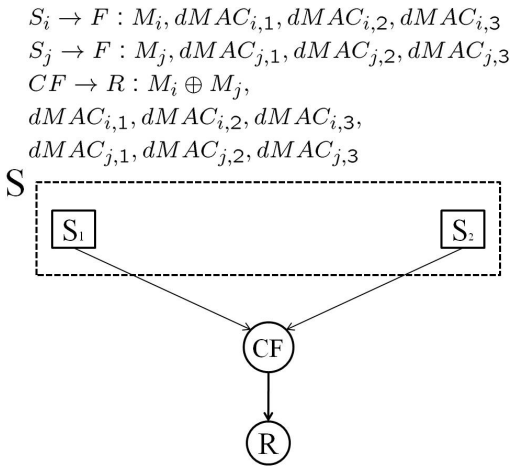

図 2: XOR ネットワーク符号化の例

して送信する例である ($t = 3$)．この例では送信ノードから 3 つずつ dMAC を受け取った転送ノード CF は，その後 6 個の dMAC をブロードキャストする．

3.2.3 dMAC の検証

送信ノードから送付データを受け取った転送ノードは受信データ中にある $id(k_{s,j})(j = 1, \ldots, t)$ を確認する．一致する $k_{s,j}$ が存在するならば, 該当する $Dec_{k_{s,j}}(dMAC_{i,j})$ を実施し, $id(k_{s,j}), r_j, h_{i,j}$ を得る．最初に復号に用いた $k_{s,j}$ のインデックスが復号した中身の $id(k_{s,j})$ と一致するか確認する．次に式 (3) を用いて r_j から h を構成する符号語の場所を決定し，h を用いて生成した $h_{i,j}$ とメッセージから得た $h_{i,j}$ と一致するか確認する．

図 2 の例で考える．$dMAC$ の入力のうち符号語を u ($u = 3$) 回実行した結果の各符号語のインデックスを各々x, y, z とする．このとき，$M_i \oplus M_j$ の符号語を，$e_x = m_{i,x} \oplus m_{j,x}$, $e_y = m_{i,y} \oplus m_{j,y}$, $e_z = m_{i,z} \oplus m_{j,z}$ とする．また，$dMAC_{i,\ell}$, $dMAC_{j,\ell}$ の復号に成功し，$h_{i,\ell} = m_{i,x} \oplus m_{i,y} \oplus m_{i,z}$ 及び $h_{j,\ell} = m_{j,x} \oplus m_{j,y} \oplus m_{j,z}$, $(\ell \in 1, 2, 3)$ を得たとする．このとき，M_i と M_j が改ざんされていないか確認する式を以下に示す．

$$e_x \oplus e_y \oplus e_z = h_{i,\ell} \oplus h_{j,\ell} \qquad (5)$$

3.3 既存研究の比較

ここでは Yu らの方式と Apavatjrut らの方式を計算量, 通信量, メモリ量, MAC の構成要素の各項目で比較する (計算量は 1 ノードあたりである)．表 1 に比較結果を示す．

表 1 における t' は復号が可能な鍵の数であり，c は転送ノードにデータを送信するノードの数であり，$1 \leq t' \leq t$, $c \geq 2$ である．計算量で XOR よりも MAC や Dec の計算量が支配的であるので, MAC で主に比較する．結果より $t' \ll c$ であるならば Yu らの方式が有利であり, そうでないならば Apavatjrut らの方式が有利になる．一方，通信量は計算量と同様 $t \ll c$ であるならば Yu らの方式が有利であり, そうでないならば Apavatjrut らの方式が有利になる．メモリ量は Apavatjrut らの方式に対して Yu らの方式が t の数に応じて大きくなる．

3.4 既存研究の課題

Apavatjrut らの方式は，データ完全性実現に使用する MAC 同士をアグリゲーションすることで MAC の数を一定に保つことが可能である．また，全ノードで 1 つの鍵を共有することによって転送ノードが MAC の検証が可能である．そのため，一見この方式は効率的に MAC をアグリゲーションできるだけでなく, 汚染されたメッセージを除去できるように思える．しかし，1 個の鍵のみで全てのデータを検証しているため, 転送ノードによるデータの偽造を防ぐことができず汚染が拡大するという問題点がある．

図 3 は送信ノード S が受信ノード R にそれぞれ 2 つのメッセージ M_1, M_2 を送信する時の例である．途中にあるネットワーク符号化を行う転送ノード CF が M_1, M_2 を受け取ったにもかかわらず, まったく関係のないメッセージ M_3 をブロードキャストし（汚染し），最終的に汚染が受信者 R にまで及んでいることがわかる．

これに対して Yu らの方式は，1 つのメッセージに対して複数の異なる MAC を使用することで転送ノードによる MAC の偽造を防いでいる．しかし，ネットワーク符号化を行う転送ノードが MAC のアグリゲーションを行わないため,

表 1: ネットワーク符号化を行う転送ノードにおける効率性比較

	計算量	通信量 (送受信)	メモリ量	MAC の構成要素
[1]	$\frac{c(c-1)}{2}(XOR_{\|M\|}+XOR_{\|MAC\|}+MAC)$	$\frac{c(c+1)}{2}(\|M\|+\|MAC\|)$	$\|k\|$	UHFs+ストリーム暗号
[3]	$t'c(u-1)(f+XOR_{\frac{\|M\|}{m}})+t'cDec+(c-1)XOR_{\|M\|}$	$t(c+2)(\|dMAC\|+\|id\|)+(c+1)\|M\|$	$t\|k\|$	共通鍵暗号

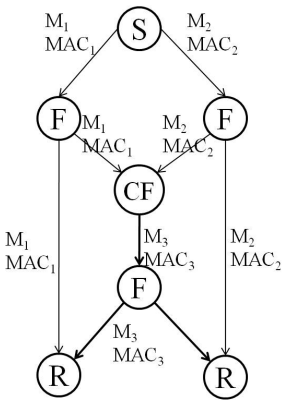

図 3: Apavatjrut らの方式における偽造の例

符号化処理が進むにつれて MAC の数が増大する課題が残る．

4 Yu らの方式の単純改良

4.1 改良点

Apavatjrut らの方式では MAC の偽造が可能であることが問題であり，Yu らの方式では転送ノードが MAC のアグリゲーションを行えないことが課題であった．そこで本節では，MAC の偽造に耐性を持つ Yu らの方式において，MAC のアグリゲーションに関する単純な改良を行った．

Yu らの方式では，単一送信ノードを想定し，各メッセージ M_i に対して t 個の dMAC が付加される．このとき，送信ノードは dMAC に対応する t 個の鍵を持っているため，ネットワーク上を流れる全ての dMAC に対して，ℓ 番目の dMAC が鍵 $k_{s,\ell}$ に対応する．また，送信ノードとは独立に t 個の鍵が転送ノードにも配布される．もし転送ノードの鍵のいくつかが送信ノードの鍵と一致するならば，転送ノードはその鍵で生成された dMAC を復号することができ，M_i の特定部分を検証できる．

では，通信量を削減するために，この dMAC のアグリゲーションを考える．Yu らの方式では，転送ノードは送信ノードの鍵と同じ鍵で生成された dMAC を復号できた．つまり，復号によってネットワーク符号化前のハッシュ値を取得できるため，そのハッシュ値を XOR することで dMAC をアグリゲーションできる．ここで，転送ノードが i 個のメッセージ $M_1,\ldots,M_i$ を受信し，ネットワーク符号化を行い $M_j = M_1\oplus,\ldots,\oplus M_i$ を次のノードへブロードキャストすることを考える．このとき，転送ノードが鍵 $k_{s,\ell}$ を持っている場合，次のように ℓ 番目の dMAC のアグリゲーションを行う．

$$dMAC_{j,\ell} = AGG_{k_{s,\ell}}(dMAC_{1,\ell},\ldots,dMAC_{i,\ell}) \quad (6)$$

ただし，アグリゲーションを行う関数 AGG は鍵 $k_{s,\ell}$ を用いて複数の dMAC を 1 つの dMAC に集約する関数であり，実質的には 1 回の dMAC の演算である．例えば，ある転送ノートが 2 つのノードから M_1, $dMAC_{1,1}$, $dMAC_{1,2}$, $dMAC_{1,3}$ と M_2, $dMAC_{2,1}$, $dMAC_{2,2}$, $dMAC_{2,3}$ を受信したとする．仮に転送ノードが鍵 $k_{s,1}$ 及び $k_{s,2}$ を持っているなら，上位ノードに対して $M_j = M_1 \oplus M_2$, $dMAC_{j,1}$, $dMAC_{j,2}$, $dMAC_{1,3}$ 及び $dMAC_{2,3}$ をブロードキャストすることになる．このとき，dMAC の数は 6 個から 4 個に削減されていることが分かる．

4.2 MAC アグリゲーションの評価

MAC がアグリゲーションされない場合，最終的な受信ノード（図 4 の R）の受信データサイズが最も大きくなると考えられる．そこで，本評価では MAC のアグリゲーションによって R が受信する dMAC 数をどの程度削減できるかを評価する．

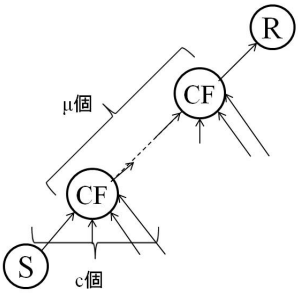

図 4: ネットワーク符号化を行う一経路のモデル

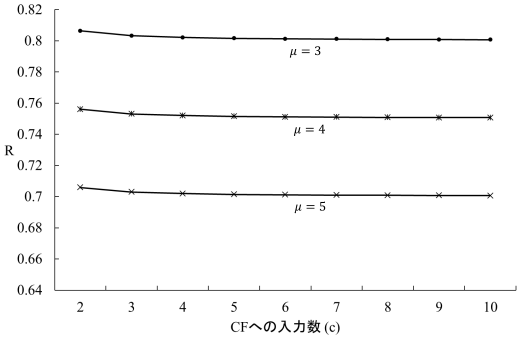

図 5: ネットワーク符号化を行う一経路のモデル

一般的に送信ノードから受信ノードへの経路は複数存在する．図 4 は複数の経路のうちの一経路を表している．ここでは，S と R の間にネットワーク符号化を行う μ 個の転送ノード (CF) が存在することを想定する．また，c を CF への入力数，v を 2 ノード間で重なる鍵数とする．このとき，dMAC のアグリゲーションによる一経路において削減される総 MAC サイズの割合の下限 R は次のように与えられる．

$$R = \begin{cases} \dfrac{(t-\mu v)((c-1)\mu+1)+v\left(\sum_{x=1}^{\mu-1}(c-1)x+\mu\right)}{t((c-1)\mu+1)} & (t \geq \mu v) \\ \dfrac{v\sum_{x=1}^{\lfloor \frac{t}{v} \rfloor -1}(c-1)x+t}{t((c-1)\mu+1)} & (t < \mu v) \end{cases} \quad (1)$$

分母の $t(c\mu+1)$ は，一経路においてアグリゲーションされない場合の R が受信する全ての dMAC の数を表す．ここで，Yu らの方式での評価と同じパラメータとして $t = 10, v = 1$ を用いる．このとき，図 5 は c と μ を変化させたときの R の変化を示している．この結果より，CF への入力数及びネットワーク符号化を行う転送ノード数が増えると，一経路における総 dMAC サイズを 7〜8 割程度に削減できることが分かった．ただし，各 CF が dMAC のアグリゲーションを行う場合，1 回の dMAC 処理が増えるのみである．

5 まとめ

今回は XOR ネットワーク符号化のうち, Yu らの提案した方式と Apavatjrut らの提案した方式を計算量や通信量, メモリ量などの観点から比較した．また，Yu らの方式に MAC 同士をアグリゲートする改良を行った．さらに，CF への入力数及びネットワーク符号化を行う転送ノード数に応じてアグリゲートできる割合がどのように変化するのかの実験も行った．今後は，汚染攻撃に対して耐性を持ちつつ, アグリゲーションによる MAC の削減率をさらに高める方式を検討する．

参考文献

[1] Anya Apavatjrut, Wassim Znaidi, Antoine Fraboulet, Claire Goursaud, Cedric Lauradoux and Marine Minier, **Energy Friendly Integrity for Network Coding in Wireless Sensor Networks**, *NSS2010*, pp.223-230.

[2] G. Brassard, **On Computationally Secure Authentication Tags Requiring Short Secret Shared Keys**, *Crypto1982*, 1982, pp.79-86.

[3] Zhen Yu, Yawen Wei, Bhuvaneswari Ramkumar and Yong Guan, **An Efficient Scheme for Securing XOR Network Coding against Pollution Attack**, *INFOCOM2009*, pp.406-414, 2009