VANETにおける経路認証について

双紙 正和 †　　三吉 雄大 †

† 広島市立大学 情報科学部
〒 731-3194 広島県広島市安佐南区大塚東三丁目 4 番 1 号
soshi@hiroshima-cu.ac.jp,miyoshi@sos.info.hiroshima-cu.ac.jp

あらまし 近年の情報関連技術の進歩により，車々間ネットワークから構成される高度な情報化交通ネットワークの実現が可能となってきた．このようなネットワークは VANET (Vehicular Ad hoc NETworks) と呼ばれ，そのセキュリティ確保が急務となっている．本論文では，特に，その効率のよい経路認証を実現するために，ハッシュ連鎖を応用した，One Way Cross Networks (OWCN) に基づいた手法を提案する．

On path authentication for VANETs

Masakazu Soshi†　　Yuudai Miyoshi†

†Faculty of Information Sciences, Hiroshima City University
3-4-1 Ozuka-Higashi, Asa-Minami-Ku, Hiroshima, 731-3194, JAPAN
soshi@hiroshima-cu.ac.jp,miyoshi@sos.info.hiroshima-cu.ac.jp

Abstract In this paper we discuss path authentication protocols for VANETs (Vehicular Ad hoc NETworks). In particular we propose an efficient protocol based on the novel concept, i.e., One Way Cross Networks (OWCN).

1 はじめに

近年，情報関連技術の進展及び情報デバイスの低価格高機能化により，自動車に各種コンピュータや組み込み機器を搭載した上でそれらの車内ネットワークを構築し，さらに，車外への無線通信を可能にするといった，高度な情報化交通ネットワークの実現が可能となってきた．こういった自動車によるネットワークは，アドホックネットワークの一種とみなすことができ，特に，Vehicular Ad hoc NETwork (VANET) と呼ばれている．

このような VANET においてもセキュリティの確保が重要となるのは言うまでもないが，VANET においては，それに特化したセキュリティ上の問題点がさまざまに指摘されており [3]，それらの解決が急務となっている．

本論文は，VANET におけるセキュリティの中でも，特に経路認証を対象とし，効率の良い認証プロトコルを提案することを目的とする．そのために，本論文では，Joye らによって提案された One Way Cross Trees (OWCT) [4] に基づいた，One Way Cross Networks (OWCN) を考案し，それに基づいた経路認証プロトコルを提案する．OWCN はハッシュ連鎖の高度な応用であり，効率よく経路認証を実現することができる．さらに本論文では，提案認証プロトコルのセキュリティを高めるため，デュアル OWCN という新たなハッシュ連鎖の構成を考え，それを利用した検証方法を議論する．本論文で提案する手法は，VANET などの ad hoc network にとどまらず，より一般に，センサーネットワー

クなどにも適した，ハッシュ連鎖ベースの効率のよい認証プロトコルとして，多くの応用を期待できる．

本論文は，以下のように構成される．まず，2節で，本研究の要素技術や従来研究について概説する．次に，3節で，One Way Cross Networks に基づいたプロトコルを提案し，4節で提案プロトコルの問題点を解決する方法を議論する．最後に5節で本論文の結論を述べる．

2 関連研究

この節では，提案方式に必要になる要素技術として，ハッシュ連鎖について述べ，次に，関連研究について簡単に議論する．

2.1 ハッシュ連鎖

本論文では，単純のため，以下の二つの性質を満たす関数 h をハッシュ関数と定義する [7]：

- 一方向性： ハッシュ値 v が与えられたとき，$v = h(x)$ を満たすような x を求めることが困難である，
- 衝突困難性： $h(x) = h(y)$ となるような，x, y $(x \neq y)$ を求めることが困難である．

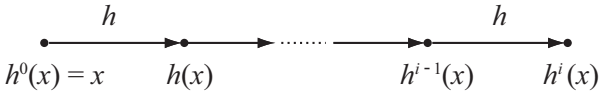

図 1: ハッシュ連鎖

ハッシュ連鎖 (hash chain) とは，入力 x にハッシュ関数 h を繰り返し適用して得られるハッシュ値の列である [5]．ここで，$h^0(x) := x$，$i > 0$ について $h^i(x) := h(h^{i-1}(x))$ とする．ハッシュ連鎖を図的に示すと，図 1 のようになる．

ハッシュ関数の性質より，$i < j$ のとき，$h^i(x)$ の値を知っていれば，$h^j(x) = h^{j-i}(h^i(x))$ のように計算することにより $h^j(x)$ を得られるが，$h^j(x)$ から $h^i(x)$ を計算することはできない．なお，本論文の以降では，ハッシュ連鎖の入力 x を種 (seed) ともいう．

2.2 ハッシュ連鎖を用いたプロトコル

Lamport は，ハッシュ連鎖を用いた認証プロトコルを提案した [5]．このプロトコルでは，ユーザ U は，初期値 s_0 を選び，ハッシュ関数 h を用いて $s_i := h(s_{i-1})$ としてハッシュ連鎖を計算する．そして，s_n を安全な方法でサーバ S に送る．最初のログイン時に，U は，s_{n-1} を S に送る．S は，$s_n = h(s_{n-1})$ が成立すれば，U のログインを許可する．2.1 節で述べたハッシュ関数の性質により，この式を満たす s_{n-1} を提示できるのは，U のみである．ログインが成功したとき，S は，s_{n-1} を保存する．次に U がログインするときには，U は s_{n-2} を S に送る．S は，$s_{n-1} = h(s_{n-2})$ が成立するかどうか確かめる．このようにして，n 回のログインが可能になる．このプロトコルの大きな特徴は，ログインのための情報 s_i は一回限りしか使えないということである．そこで，s_i はその平文のまま送ることができ，再送攻撃も適用できない．

2.3 VANET における認証プロトコル

VANET におけるセキュリティについては近年さまざまな研究がなされているが，いまだにまとまったものはない．この節では，その中でも，本研究に関連付けられそうなものとして，Hsiao らによる，Fast Authentication (FastAuth) プロトコル [2] について議論する．

FastAuth は，VANET における signature flooding に対処するため，効率の良いブロードキャスト認証プロトコルを実現しようとするものである．そのために，車の進行方向を右，左，直進の三方向とし，それぞれについて進行確率を計算してハフマン符号化し，そのハッシュ木を計算する．この手法自体は興味深いものであるが，車の進行方向の確率を計算することは困難であり，実用性には疑問が残る．一方，我々の提案手法では，そのような問題はない．

3 提案手法

この節では，車が道路を走行中のとき，その経路を他の実体に対して認証する方法を提案す

る．VANET では，複数の車が高速に移動するため，認証の遅延や，攻撃者による攻撃や妨害が，大規模な事故につながる可能性が生じる．したがって，特に VANET においては，高速な認証法が極めて重要である．特にこの論文では，車の移動方向を認証できる方法を考える．

そこで本論文では，Joye らが提案した One-Way Cross-Trees (OWCT と略す) [4] をベースにした，効率の良い認証方法を提案する．

3.1 前提

本論文では，それぞれの車は，権威ある certification authorities (CAs) によって作成された公開鍵を持つ，署名機能を利用できるものと仮定する [8, 10]．しかしながら，本研究では，効率の良い認証プロトコルを構築するため，できる限り署名機能は使わないことを目指す．

次に，本論文で提案する One Way Cross Networks (OWCN) の定義を，定義 1 に与える．

定義 1 (k-OWCN $(m_1, m_2, \ldots, m_k)$)**.** h_1, h_2, ..., h_k を k 個のハッシュ関数とし，($s_{1,0}$, $s_{2,0}$, ..., $s_{k,0}$) を，種の k 個組とする．k-OWCN (m_1, m_2, ..., m_k) とは，

$$V_{r_1,r_2,\ldots,r_k} := (h_1^{r_1}(s_{1,0}), h_2^{r_2}(s_{2,0}), \ldots, h_k^{r_k}(s_{k,0})) \tag{1}$$

を頂点とする有向グラフである．ただし，$0 \leq r_i \leq m_i$ $(i = 1, \ldots, k)$ とする．ここで，ある $V_{r'_1,r'_2,\ldots,r'_k} = (h_1^{r'_1}(s_{1,0}), h_2^{r'_2}(s_{2,0}), \ldots, h_k^{r'_k}(s_{k,0}))$ について，あるただ一つの i $(1 \leq i \leq k)$ が存在して，$r'_i = r_i + 1$ であり，かつ，それ以外の j $(\neq i)$ のときには $r'_j = r_j$ となるとき，その時に限り，$V_{r_1,r_2,\ldots,r_k}$ から $V_{r'_1,r'_2,\ldots,r'_k}$ への辺が存在する．

例として，3-OWCN $(2, 2, 2)$ を図 2 に示す．ただし，図 2 においては，簡単のため式 (1) の頂点を $(r_1, r_2, \ldots, r_k)$ などと書いている．

以上から明らかなように，k-OWCN (m_1, m_2, ..., m_k) とは，Joye らの OWCT において，k 個のハッシュ関数のそれぞれの適用回数に m_1, m_2, ..., m_k という上限を与えたものである．

次に，k-OWCN $(m_1, m_2, \ldots, m_k)$ の背景および概要について述べる．k-OWCN (m_1, m_2, ..., m_k) においては，時間を離散的なタイムスロットで考え，それぞれのスロットで，車がある一つの方向に進んでいるような状況を認証する．より詳しくは，ある時点における車の移動方向が，$D_1, D_2, \ldots, D_k$ という，最大 k 通りあると考える．さらに，それぞれの方向 D_i を，それぞれ最大 m_i 回とることができるものとする $(i = 1, 2, \ldots, k)$.

また，Joye らの OWCT と我々の OWCN の違いに注意せよ．OWCT と異なり，OWCN は木にならず，有向グラフとなる．また，OWCT は，木の葉が複数存在するため，それを使った効率の良い認証は困難である．一方，OWCN は，すべての頂点が到達可能な $V_{m_1,m_2,\ldots,m_k}$ がただ一つ存在し，これが，以降で述べるように，ハッシュ連鎖ベースの効率的な認証を可能とする．

3.2 提案プロトコル

以上のような議論のもとに，この節では k-OWCN $(m_1, m_2, \ldots, m_k)$ に基づいた，VANET における経路認証プロトコルを提案する．

以下に，提案プロトコルを示す．ここでは，最大 r 回認証を行うものとする $(r \geq 1)$.

1. 証明者 (車) は，状況に応じた k, m_1, m_2, ..., m_k を選び，k-OWCN (m_1, m_2, ..., m_k) を構築する．
2. 証明者は，$(v_{1,r}, v_{2,r}, \ldots, v_{k,r}) := V_{m_1,m_2,\ldots,m_k}$ を自らの署名付きで検証者に送る．
3. 検証者は，署名を検証し，正しければ ($v_{1,r}$, $v_{2,r}$, ..., $v_{k,r}$) を保存する．
4. ℓ 回目の認証ステップは以下のようになる $(\ell \geq 1)$.
   (a) 検証者は ($v_{1,r-\ell+1}$, $v_{2,r-\ell+1}$, ..., $v_{k,r-\ell+1}$) $(= V_{\alpha_1,\alpha_2,\ldots,\alpha_k}$ と仮定する) を保存している．特に $\ell = 1$ のとき，$\alpha_i = m_i$, $v_{i,r} = h_i^{m_i}(s_{i,0})$ である．

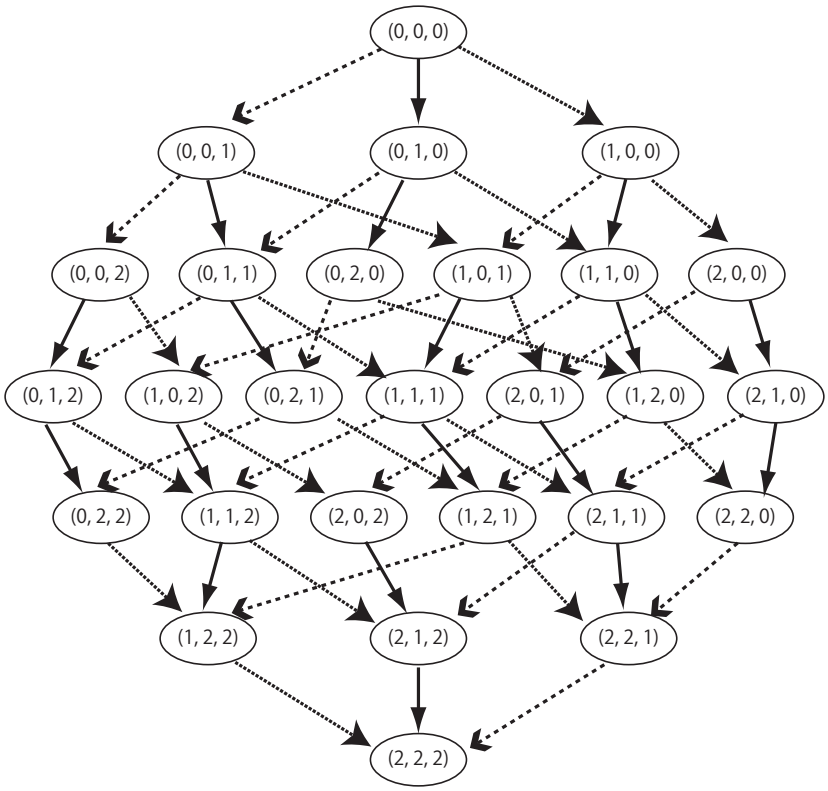

図 2: 3-OWCN $(2, 2, 2)$

(b) 証明者は，ある方向 D_i に進もうとするとき，$(v_{1,r-\ell}, v_{2,r-\ell}, \ldots, v_{k,r-\ell}) := V_{\alpha_1,\alpha_2,\ldots,\alpha_i-1,\ldots,\alpha_k}$ として，$v_{i,r-\ell}$ のみ検証者に送る．ただし，$\alpha_i \geq 1$ でなければならない．

(c) 検証者は，$v_{i,r-\ell+1} = h_i(v_{i,r-\ell})$ が成立するかどうか確かめる．ハッシュ関数の一方向性より，このような $v_{i,r-\ell}$ を計算できるのは正しい証明者のみである．よって，成立するとき，方向 D_i が認証されたものとし，$v_{i,r-\ell+1}$ に $v_{i,r-\ell}$ を上書きする．こうして検証者に保存されているものを，$(v_{1,r-\ell}, v_{2,r-\ell}, \ldots, v_{k,r-\ell})$ とする．ただし，i については $h_i(v_{i,r-\ell}) = v_{i,r-\ell+1}$ が成立し，それ以外の j $(\neq i)$ について，$v_{j,r-\ell} = v_{j,r-\ell+1}$ である．

3.2.1 プロトコルの実施例

ここでは提案プロトコルの実施例として，図 2 の 3-OWCN $(2, 2, 2)$ $\mathcal{T}$ について考えてみる．

$k = 3$ であるから，$\mathcal{T}$ では，3 方向 $D_1 = \mathrm{L}$ (left), $D_2 = \mathrm{F}$ (forward), $D_3 = \mathrm{R}$ (right) を考えることができる．さらに，$(m_1, m_2, m_3) = (2, 2, 2)$ であるから，L, F, R のそれぞれの方向を高々2 回づつ進むことができる．

このとき，車 (証明者) が，$\mathrm{L} \to \mathrm{F} \to \mathrm{R} \to \mathrm{L} \to \mathrm{F} \to \mathrm{R}$ の順で進んだとしよう．簡単のため式 (1) の頂点を $(r_1, r_2, \ldots, r_k)$ などと書くとすると，このとき，車は順に $(1, 2, 2)$, $(1, 1, 2)$, $(1, 1, 1)$, $(0, 1, 1)$, $(0, 0, 1)$, $(0, 0, 0)$ を提示していけばよい (なお，言うまでもなく，実際にそれぞれのステップで提示するのは，一つのハッシュ値だけである)．

以下では，便利のため，上記の経路を L (1, 2, 2) → F (1, 1, 2) → R (1, 1, 1) → L (0, 1, 1) → F (0, 0, 1) → R (0, 0, 0) などと書いたり，あるいはさらに簡単に，(1, 2, 2) → (1, 1, 2) → (1, 1, 1) → (0, 1, 1) → (0, 0, 1) → (0, 0,

0) などと書いたりすることがある．

3.2.2 議論

最初に，本提案プロトコルが非常に効率が良いものであることに注意されたい．プロトコルの開始時にのみ，OWCN を計算し，さらに，k 個のハッシュ値 $(v_{1,r},\ v_{2,r},\ \ldots,\ v_{k,r}) = V_{m_1,m_2,\ldots,m_k}$ およびその署名を検証者に送り，その署名を検証しなければならないものの，以降では，一つのハッシュ値 $v_{i,r-\ell}$ のみ送り，そのハッシュ関数の値のみを計算すればよい．したがって，プロトコル実施時の通信量および計算量は，2.2 節で述べた，ハッシュ連鎖を用いたプロトコル (いわゆる one time password) と同等であり，非常に効率が良い．

また，一般的にいって VANET ではパケット紛失率が高いと考えられるが [2]，TESLA と同様の理由で，本提案プロトコルも，パケット紛失に対して頑健であると考えられる．たとえば，3.2.1 節の例を再び考えると，仮に (1, 1, 2) の値が紛失したとしても，以降の (1, 1, 1) などから再計算することが可能である．

4 デュアル OWCN に基づく検証

この節では，3.2 節で提案したプロトコルに対する攻撃と，それへの対策について議論する．

4.1 なりすまし攻撃

図 3 のように，車 A, B, C が，右方向に走行しているような状況を考える．ここで，A, B, C が，それぞれ証明者，攻撃者，検証者であり，提案プロトコルを実施していると仮定する．

まず，当初は A の無線範囲に B, C が存在したとしよう．ところが，C が高速に移動したため，A の無線範囲から外れてしまったが，一方で B は A の無線範囲に残ったとする．さらに，C は B の無線範囲に存在するとする．このような状況が，図 3 にまとめられている．

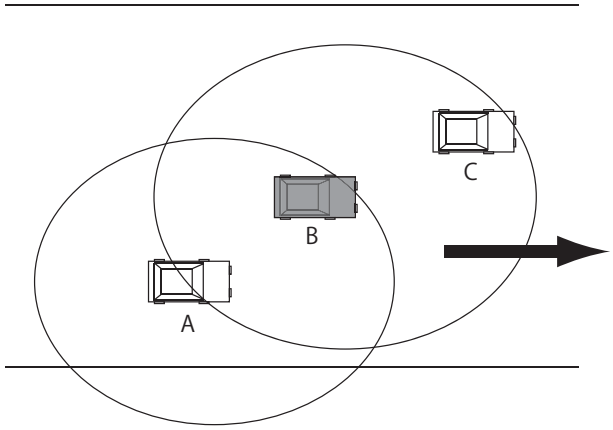

図 3: なりすまし攻撃

このとき，以下のような攻撃が可能になる[1]．定義 1 より，k-OWCN $(m_1,\ m_2,\ \ldots,\ m_k)$ においては，$V_{r_1,r_2,\ldots,r_k}$ が公開されれば，任意の $V_{r'_1,r'_2,\ldots,r'_k}$ (ただし，$r'_i \geq r_i,\ 1 \leq i \leq k$) を計算可能であることに注意せよ．そして，3.2.1 節であげた例を図 3 に適用して考える．すなわち，車 A (証明者) は L $\rightarrow$ F $\rightarrow$ R $\rightarrow$ L $\rightarrow$ F $\rightarrow$ R の順で進み，車 C (検証者) に対して，3-OWCN $(2,2,2)$ $\mathcal{T}$ に従って，順に $(1,2,2)$, $(1,1,2)$, $(1,1,1)$, $(0,1,1)$, $(0,0,1)$, $(0,0,0)$ を提示しようとしていると仮定する．

このとき，車 B は，A からのメッセージをすべて受信し，そのまま C に転送せずに，いったんバッファリングして改ざんしてから転送することが可能となる．たとえば，B が $(1,1,1)$ を受信した段階で，B は，(r'_1, r'_2, r'_3) (ただし，$r'_i \geq 1,\ i = 1,2,3$) を計算できるようになる．そこでたとえば，A の移動が，R $(2,2,1)$ $\rightarrow$ L $(1,2,1)$ $\rightarrow$ L $(1,1,1)$ $\cdots$ であるかのように偽装することが可能となってしまう．

これは，TESLA [9] (正確には TESLA Scheme I: The Basic Scheme) や TSVC [6] 等，ハッシュ連鎖を用いたプロトコルには本質的に存在する問題点であると考えられる．これを解決するために，たとえば TESLA では，パケットの到着時間に制約を加える等の解決策をとっているが，このような時間制約を VANET のような環境で適切に設定することは容易ではない．

[1] TSVC [6] に対する本質的には同等の攻撃が指摘されている [1].

4.2 デュアル OWCN

本論文では，4.1 節で述べたような問題の解決策に向けて，デュアル OWCN の使用を提案する．

定義 2 (デュアル k-OWCN $(m_1, m_2, \ldots, m_k)$)**.** $h_1, h_2, \ldots, h_k$ を，ハッシュ関数とし，$(s_{1,0}, s_{2,0}, \ldots, s_{k,0})$ を，種の k 個組とする．デュアル k-OWCN $(m_1, m_2, \ldots, m_k)$ とは，

$$V_{r_1,r_2,\ldots,r_k} := (h_1^{m_1-r_1}(s_{1,0}), h_2^{m_2-r_2}(s_{2,0}), \ldots, h_k^{m_k-r_k}(s_{k,0})) \quad (2)$$

を頂点とする有向グラフである．ただし，$0 \leq r_i \leq m_i$ $(i = 1, \ldots, k)$ とする．ここで，ある $V_{r'_1,r'_2,\ldots,r'_k} = (h_1^{m_1-r'_1}(s_{1,0}), \ldots, h_k^{m_k-r'_k}(s_{k,0}))$ について，あるただ一つの i $(1 \leq i \leq k)$ が存在して，$r'_i = r_i - 1$ であり，それ以外の j $(\neq i)$ のときには $r'_j = r_j$ となるとき，その時に限り，$V_{r_1,r_2,\ldots,r_k}$ から $V_{r'_1,r'_2,\ldots,r'_k}$ への辺が存在する．

4.2.1 例

デュアル OWCN の例を，図 4 に示す．図 4 (a) が，2-OWCN (2, 2) であり，そのデュアルとなるのが，図 4 (b) のデュアル 2-OWCN (2, 2) である．

4.2.2 デュアル OWCN による経路範囲制限

この節では，デュアル OWCN を用いた，経路情報の改ざん対策を述べる．

例として，図 3 のように，車 A, B, C が道路を走行しており，A, B, C が，それぞれ証明者，攻撃者，検証者であるような状況を考える．さらに，単純のため，図 4 の OWCN を用いて，経路認証を行っているとする．このとき，A (証明者) は，図 4 (a) の 2-OWCN (2, 2) $\mathcal{T}$ を構成したとする．また，経路確認者 (4.3 節で議論する) は，図 4 (b) のデュアル 2-OWCN (2, 2) $\mathcal{T}'$ を構成し，$\mathcal{T}'$ の頂点 $V_{0,0}$ を署名付きで C (検証者) に公開する．C は，その署名が正しければ ($\mathcal{T}$ の $V_{2,2}$ とともに) $\mathcal{T}'$ の $V_{0,0}$ を保存しておく．$\mathcal{T}, \mathcal{T}'$ ともに，取り得る方向は 2 通りであり，$D_1 = \mathrm{L}, D_2 = \mathrm{R}$ としておく．以下では，以前のように，頂点 V_{r_1,r_2} を，(r_1, r_2) などと略記する．

このとき，A は，$\mathrm{R} \to \mathrm{R} \to \mathrm{L} \to \mathrm{L}$ と進んでいくと仮定する．したがって，A は，$\mathcal{T}$ の (2, 1), (2, 0), (1, 0), (0, 0) を C に順に提示していくことになる．

ここで，B (攻撃者) が図 3 のような状況で成りすまし攻撃を行い，経路を改ざんして C に通信したとしよう．たとえば，B は，A からの通信を (1, 0) まですべて受け取り，それから経路を改ざんし，$\mathrm{L} \to \mathrm{R} \to \mathrm{R}\ (\to \mathrm{L})$ のようになるよう，$\mathcal{T}$ の (1, 2), (1, 1), (1, 0) を順に C に送信したとする．

このような状況で，例として，経路確認者は，A が $\mathcal{T}$ の (1, 0) を送信した後，過去走行してきた経路の範囲を認証するために，デュアル OWCN $\mathcal{T}'$ の頂点 (2,1) を送信したとする．以上のような状況を図 5 に与える．

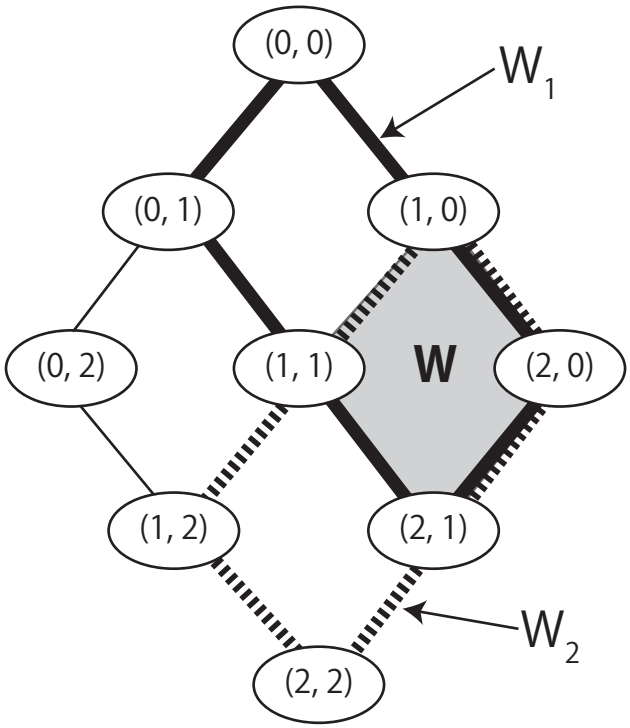

図 5: デュアル OWCN による範囲制限

図 5 は，以下のように解釈される．A が $\mathcal{T}$ の (1, 0) を提示した時点で，$\mathcal{T}$ の頂点で生成可能なものは，図 5 の太い点線で囲まれた部分 W_2 の頂点である (よって，攻撃者 B は，(1, 2), (1,

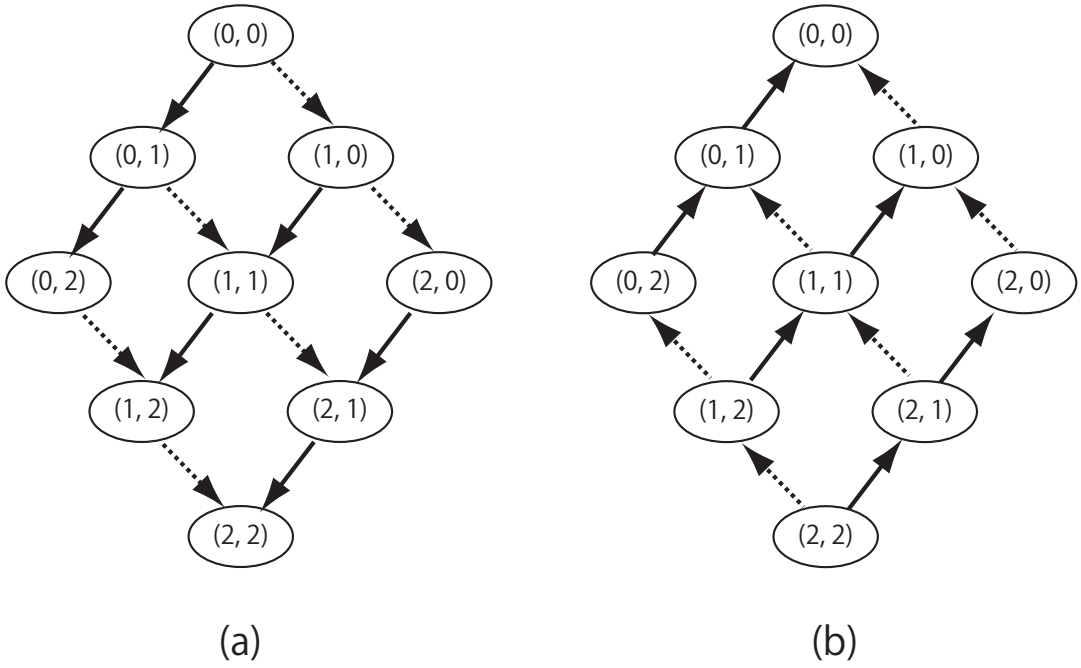

図 4: (a) OWCN と (b) デュアル OWCN

1), (1, 0) という頂点を公開し，経路を改ざんできた)．ここで次に，経路確認者は $\mathcal{T}'$ の (2, 1) を公開することで，経路範囲の制限を表現しようとする．すなわち，$\mathcal{T}'$ において，(2, 1) を公開することで計算可能な頂点は，図 5 の実線の黒い太線で囲まれた部分 W_1 に属する頂点である．このような W_1 と W_2 の共通部分 W における経路を，走行可能であった経路として解釈する．したがって，この場合は，可能性としては (2, 1) → (2, 0) → (1, 0) か，(2, 1) → (1, 1) → (1, 0) のいずれかということになる．そして，以上の二つの経路とも，(1, 2) を通ることはありえないので，B による経路改ざんが発覚する．

4.3 議論

4.2.2 節で述べた検証手法において，経路確認者を担当する実体として最も望ましいのは，Road Side Unit (RSU) [8, 10] であると考えられる．一般的に言って，(複数の) RSU は車 A, C と通信できるためである．なお，提案手法は，デュアル OWCN の最初の構築を除けば，4.2.2 節で述べた経路確認自体は負荷をかけることなく，効率よく実現できることに注意せよ．

経路確認者を，証明者 (車 A) が担当することも考えられる．しかしこの場合，攻撃者 B がその経路確認メッセージを検証者 C に中継しないこともあり得るので，タイムアウトなどを導入して，経路確認メッセージが受信できない場合は，経路認証プロセス全体を破棄する，といった対策が必要になってくる．

また，4.2.2 節で述べた検証手法において重要となるのは，経路確認者がデュアル OWCN のいずれの頂点を公開するかという点である．たとえば，4.2.2 節の例において，もし証明者が $\mathcal{T}'$ の (2, 1) のかわりに (2, 2) を公開したとすると，結果として (1, 2) も経路範囲 W_1 に含まれることになり，その改ざんを検出できなくなってしまう．

一般的に言えば，経路確認者は，W が，許容できる最も広い範囲となるように $\mathcal{T}'$ の頂点を公開すべきである．たとえば，図 5 においては，真の経路は (2, 1) → (2, 0) → (1, 0) であるが，仮に，攻撃者 B の改ざんによってそれが (2, 1) → (1, 1) → (1, 0) と検証者に理解されたとしても，許容できる範囲の経路であると判断した場合に，$\mathcal{T}'$ の (2, 1) を公開すべきである．

もちろん，1 個だけではなく $\mathcal{T}'$ の複数の頂点を提示することも有効であるが，それは性能とのトレードオフとなる．

5 結論

本論文では，VANET のための経路認証の実現のために，One Way Cross Networks (OWCN) に基づいたプロトコルを提案した．OWCN はハッシュ連鎖の高度な応用であり，効率よく経路認証を行うことができる．さらに本論文では，提案認証プロトコルのセキュリティを高めるため，デュアル OWCN を新たに構築し，それを利用した検証方法を議論した．本論文で提案した手法は，VANET などの ad hoc network にとどまらず，より一般に，センサーネットワークなどにも適した，ハッシュ連鎖ベースの効率のよい認証プロトコルとして広い適用範囲を期待できる．

参考文献

[1] M. Burmester, E. Magkos, and V. Chrissikopoulos. Secure and privacy-preserving, timed vehicular communication. *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, 2012. To appear.

[2] H.-C. Hsiao, A. Studer, C. Chen, A. Perrig, F. Bai, B. Bellur, and A. Iyer. Flooding-resilient broadcast authentication for VANETs. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom 2011)*, pp. 193–204, 2011.

[3] IPA 独立行政法人 情報処理推進機構. 2010 年度 自動車の情報セキュリティ動向に関する調査報告書, Apr. 2011.

[4] M. Joye and S. Yen. One-way cross-trees and their applications. In *Public Key Cryptography (PKC)*, Vol. 2274 of *Lecture Notes in Computer Science*, pp. 346–356. Springer-Verlag, 2002.

[5] L. Lamport. Password authentication with insecure communication. *Commun. ACM*, 24(11):770–772, Nov. 1981.

[6] X. Lin, X. Sun, X. Wang, C. Zhang, P.-H. Ho, , and X. Shen. TSVC: Timed efficient and secure vehicular communications with privacy preserving. *IEEE Transactions on Wireless Communications*, 7(12):4987–4998, Dec. 2008.

[7] 岡本, 山本. 現代暗号. 情報科学の数学. 産業図書, 1997.

[8] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux. Secure vehicular communication systems: Design and architecture. *IEEE Communications Magazine*, 46(11):100–109, Nov. 2008.

[9] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. Efficient authentication and signing of multicast streams over lossy channels. In *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 56–73, May 2000.

[10] M. Raya and J.-P. Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1):39–68, Jan. 2007.