6D-7

ネットワーク符号化通信のための動的 ID ベース署名方式の VANET でのシミュレーション

山本 泰資† 毛利 公美‡ 白石 善明†

名古屋工業大学† 岐阜大学‡

1. はじめに

ITS（高度道路交通システム）の研究分野の一つに安全運転支援システムがある．安全運転支援システムの分野で求められる通信要件は厳しい．文献[1]において，例えば，右折時衝突防止のための通信要件では「80%以上のデータ受信率」（要件 1），「1 秒以内の許容遅延」（要件 2）が挙げられている．

文献[2]（以降 KFMS11 方式）により，要件 1，2 を満たす車載カメラ画像による視覚支援のためのランダムネットワーク符号化を用いた車々間通信による多対多ブロードキャスト通信方式が提案されている．

VANET（Vehicular Ad-hoc NETwork）では改ざん・偽造による汚染攻撃の脅威がある．汚染パケット数が同じとき，単純なブロードキャスト通信に比べて，通信効率を向上させるためにネットワーク符号化を適用した通信の方が，複数のパケットから復号されることから汚染パケットの影響範囲は大きくなる．

文献[3][4][5]で，ネットワーク符号化のための署名方式が提案されている．文献[4][5]は，電子署名により汚染攻撃への耐性を持つが，一定数集めた署名から署名鍵を求めることができ，すなわち，電子署名の偽造ができることになる．文献[3]は，[4][5]の方式を基に署名を生成するが，署名鍵を更新することで署名収集による署名偽造を防止する方式を示している．

本稿では，汚染攻撃，署名偽造攻撃に耐性のある文献[3]の署名方式を KFMS11 方式に適用することを考え，シミュレーション実験によって署名生成，検証コストが通信に与える影響と攻撃者ノードがネットワーク符号化通信に与える汚染攻撃の影響を評価する．

2. Jiang らの方式[3]

文献[3]の方式では署名に通信オーバヘッドが少なくてすむ双線形写像を用いた ID ベース署名方式が用いられている．また，ネットワーク符号化に用いられる線形結合に適合するように署名を生成することで，中継ノードで再符号化を行う際に符号化データを復号することなく，署名を生成することができる．

2.1. パラメータの設定

双線形写像を行うためのパラメータを設定する．素数位数である q の巡回加法群，巡回乗法群をそれぞれ $\mathbb{G}$，$\mathbb{G}_T$ とし $P\in\mathbb{G}$ とおく．$\hat{e}$ を $\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ の写像とする．$H(\cdot)$を MapToPoint ハッシュ関数とし，$H:\{0,1\}^*\to\mathbb{G}$ とする．

次に署名鍵・検証鍵を生成する．送信ノードは $m+n$ 個の秘密鍵$(s_1,s_2,\ldots,s_{m+n})$を生成，署名鍵のランダム偽造攻撃に耐性を持たせるために，署名鍵生成に用いる疑似 ID(以後 *PID*)を情報送信ノードの *ID* と一方向性ハッシュ関数 h から以下のように生成する．

$$PID=h^k(ID)=h^k(\ldots(h^1(ID))\ldots)$$

ここでの k はハッシュ関数を使用した回数である．署名鍵 ***SK*** を *PID* と秘密鍵 $s_i(1\leqq i\leqq m+n)$を用いて以下のように $m+n$ 個生成する．

$$\boldsymbol{SK}=\{SK_i|SK_i=s_iH(PID),\ 1\leqq i\leqq m+n\}$$

検証鍵 ***PK*** を秘密鍵 $s_i(1\leqq i\leqq m+n)$と P を用いて以下のように生成する．

$$\boldsymbol{PK}=\{PK_i|PK_i=s_iP,\ 1\leqq i\leqq m+n\}$$

送信ノードは公開パラメータとして{G,G$_T$,q,P,***PK***,*ID*}を公開する．

2.2. 署名生成

送信ノードは m 個のシンボルのパケットを n 個送信することとする．パケットを以下のように定義する．

$$\widetilde{B}_i=[B_i,U_i]=[\underbrace{b_{i,1},\ldots,b_{i,m}}_{m},\underbrace{0,\ldots,0}_{i-1},1,\underbrace{0,\ldots,0}_{n-i}]$$

$$=[\underbrace{\widetilde{b}_{i,1},\ldots,\widetilde{b}_{i,m}}_{m},\underbrace{\widetilde{b}_{i,m+1},\ldots,\widetilde{b}_{i,m+n}}_{n}]$$

B_iは送信データを表し，U_iはその送信データが i 番目のデータであることを示している．次に，符号化ベクトル g_i を用いた符号化パケットを以下のように定義する．

$$\widetilde{Y}=\sum_{i=1}^{n}g_i\widetilde{B}_i=\sum_{i=1}^{n}g_i\cdot[B_i,U_i]=[\underbrace{y_1,\ldots,y_m}_{m},\underbrace{g_1,\ldots,g_n}_{n}]$$

$$=[\underbrace{\widetilde{y}_1,\ldots,\widetilde{y}_m}_{m},\underbrace{\widetilde{y}_{m+1},\ldots,\widetilde{y}_{m+n}}_{n}]$$

$$y_j=\sum_{i=1}^{n}g_i\widetilde{b}_{ij}\left(1\leq j\leq m+n\right)$$

パケット $\widetilde{B}_i$ に対する署名を以下のように生成する．

$$H_s(\widetilde{B}_i)=\sum_{j=1}^{m+n}\{\widetilde{b}_{i,j}SK_j\}=\sum_{j=1}^{m+n}\{\widetilde{b}_{i,j}s_jH(PID)\}$$

同様に符号化データ $\widetilde{Y}$ に対する署名を以下のように生成する．

$$H_s(\widetilde{Y})=\sum_{j=1}^{m+n}\{\widetilde{y}_jSK_j\}=\sum_{j=1}^{m+n}\{\widetilde{y}_js_jH(PID)\}$$

$$=H_s(\sum_{i=1}^{n}g_i\widetilde{B}_i)=\sum_{i=1}^{n}g_iH_s(\widetilde{B}_i)$$

符号化データに対する署名は，元々のデータに対する署名と符号化ベクトルによる線形結合であることがわかる．つまり，パケットの符号化の際，署名生成のために符号化データを復号することなく署名を更新できる．送信・中継ノードは署名生成後に，$\{PID,k,\widetilde{B}_i,H_s(\widetilde{B}_i)\}$ もしくは $\{PID,k,\widetilde{Y},H_s(\widetilde{Y})\}$ を送信する．

2.3. 署名検証

中継，受信ノードはパケット $\{PID,k,\widetilde{B}_i,H_s(\widetilde{B}_i)\}$ もしくは $\{PID,k,\widetilde{Y},H_s(\widetilde{Y})\}$ を受け取ったとき，公開パラメータ{G,G$_T$,q,P,***PK***,*ID*}を用いて以下の検証手順を行い符号化パケットの認証を行う．

【Step1】*PID* の完全性検証

$$PID\overset{?}{=}h^k(ID)$$

【Step2】署名の完全性検証

$$\hat{e}(H_s(\widetilde{Y}),P)\overset{?}{=}\hat{e}(H(PID),\sum_{j=1}^{m+n}\{\widetilde{y}_jPK_j\})$$

3. シミュレーション実験

文献[2]の通信方式に対して，署名生成，検証コストが通信に与える影響をシミュレーション実験により確認する．

3.1. シミュレーションモデル

ネットワークシミュレータとして OMNeT++4.0[6]，シミュレータのフレームワークとして MiXiM1.1[7]を使用した．KFMS11 方式では交差点付近での動作を想定している．交差点から 200[m]以内を通信有効範囲とし，交差点から 50[m]の地点で情報を習得し，情報を通信有効範囲内で中継する．直進車両

Simulation of a Dynamic-Identity Based Signature Scheme for Network Coding in VANET

†Taisuke YAMAMOTO and Yoshiaki SHIRAISHI・Nagoya Institute of Technology

‡Masami MOHRI・Gifu University

速度を 60[km/h]，右左折車両速度を 20[km/h]．車頭距離を直進車両は 40[m]，右左折車両は 20[m]とする．

3.2. シミュレーション内容

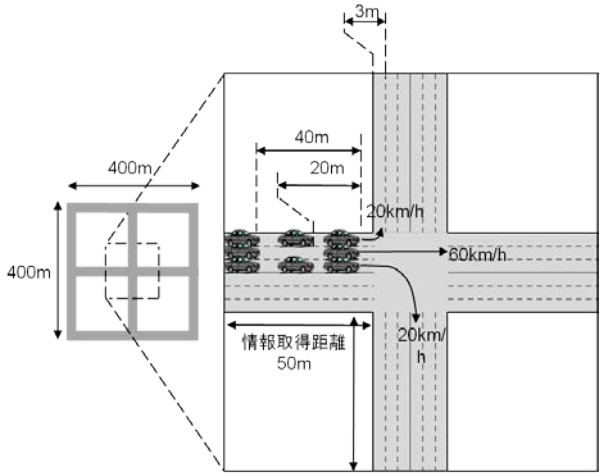

図 1　シミュレーションモデル

KFMS11 方式と KFMS11 方式に署名処理を追加したものを比較し評価した．署名検証処理はパケット受信時に行うものとし，復号や符号化に使用する符号化パケットは検証処理が完了しているパケットのみを使用する．これらの方式で，攻撃者ノードなしの状態で要件 1 の最大通信遅延と攻撃者ノードありの状態で要件 2 のデータ受信率を比較する．

攻撃者モデルを次のようにする．受信パケットの符号化データと符号化ベクトルを改ざんし送信する．送信周期を 100[ms]とする．

送信データは 1500[byte]のパケットで送信されるものとする．また，文献[3]より，署名付きの方式では符号化パケットに *PID*(44byte)と署名情報(22byte)を追加することになり，1 パケットにつき，66[byte]の通信オーバヘッドがある．要件 2 を満たすために，データの有効期限を 1000[ms]とする．送信ノードのデータ送信周期は 100[ms]，中継ノードの中継周期は 500[ms]とする．攻撃者ノードの数を 1 台とする．検証コストは 1[ms]から 100[ms]とする．

3.3. シミュレーション結果と考察

攻撃者ノードなしの状態での最大通信遅延と送信データサイズの関係を図 2 に示す．署名検証コストが 100[ms]以内であれば KFMS11 方式とほぼ同じサイズのデータを送信できることがわかる．

これは，送信周期が 100[ms]，中継周期が 500[ms]でパケット有効期限が 1000[ms]であるため，ネットワーク内で伝送されるパケット数が少なく，受信パケット数が少ないからである．また，全てにおいて約 100[ms]多く遅延が発生しているのは，通信オーバヘッドにより，送信パケット数が増えたためであると考えられる．

攻撃者ノードありの状態でのデータ受信率と送信データサイズの関係を図 3 に示す．検証を行わない場合，9000[KB]以降に受信率が低下している．9000[KB]以上のデータは 6 個以上のパケットに分割されて送信される．送信周期 100[ms]，中継周期を 500[ms]としているため，6 個目以降のパケットの多くには改ざんされたデータが符号化パケットに含まれており，周囲のノードに感染していき，復号処理を妨げていることがわかる．

4. おわりに

本稿では，シミュレーション実験によってネットワーク符号化通信を用いた車々間通信において，署名処理と攻撃者ノードが通信に与える影響を確認した．

要件 2 で与えられるようなパケットの有効期限が短く，ネットワーク上に伝送されるパケット数が限られていると，受信パケット数が少なくなるので，署名コストが 100[ms]以内であれば通信要件を満たせることを確認した．

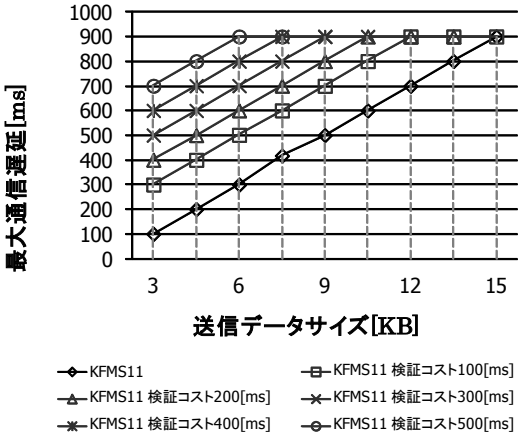

図 2　最大通信遅延

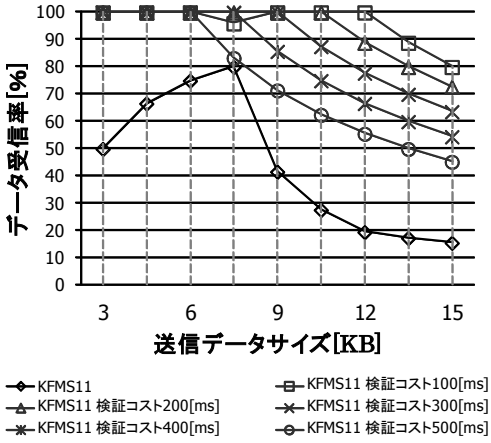

図 3　データ受信率

攻撃者ノード数や中継周期，攻撃者送信周期を変更したときの通信性能に与える影響を調べることを今後の課題とする．

参考文献

[1]総務省事務局：ITS 無線システムの高度化に関する研究会作業班（第 4 回会合）資料 4−4, “アンケートとりまとめ結果 I ～利用イメージの明確化のためのアンケート～”，入手先〈 http://www.soumu.go.jp/main_sosiki/joho_tsusin/policyreports/chousa/its/pdf/090121_2_si4-4.pdf〉(参照 2011-12-09)

[2]岸本侑大，福田洋治，毛利公美，白石善明，“車載カメラ画像による視覚支援のためのランダムネットワーク符号化を用いた車々間通信による多対多ブロードキャストの再符号化処理について”，第 73 回全国大会講演論文集 2011(1)，pp.109-110(2011).

[3]Y. Jiang, H. Zhu, M. Shi, X.(. Shen, and C. Lin, "An efficient dynamic-identity based signature scheme for secure network coding", presented at Computer Networks, 2010, pp.28-40.

[4]D. Charles, K. Jian, K. Lauter, Signature for Network Coding, Technique Report MSR-TR-2005-159, Microsoft, 2005.

[5]Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An Efficient Signature-Based Scheme for Securing Network Coding Against Pollution Attacks", in Proc. INFOCOM, 2008, pp.1409-1417.

[6]OMNeT++, http://www.omnetpp.org/

[7]MiXiM, http://sourceforge.net/projects/mixim/