岐阜大学自然科学技術研究科

知能理工学専攻

令和 5 年度学位論文

デジタル署名を用いたセキュアな V2V アドホックルーティングプロトコル

三嶋研究室

学籍番号:1224525046

階戸 弾

指導教員:三嶋 美和子 教授

目次

はじめに

VANET (Vehicle Ad hoc Network)とは，移動する車両間で通信を行うためのモバイルアドホックネットワークである．車両間の通信は直接行われるか，路側器または近隣の車両を介してマルチホップ通信が用いられる．VANET は，近年の車両技術の進歩と密接に関連しており，都市化の進展と交通密度の増加や自動運転技術がレベル 3 まで進んできていることなどに伴い，実用化への期待が高まっている．例えば，見通しの悪い交差点での出会い頭事故，右左折事故などを防止することや，リアルタイムに交通情報を提供することで，渋滞の緩和や迂回路の提案など，スムーズな交通フローを含めた自動運転の実現に貢献すると考えられている．

しかし，高く安定した通信性能と同時に，通信のプライバシーおよびセキュリティの確保なくして VANET の実用化はあり得ない．過去十年にわたり，これらの課題に対する様々な方法が研究されてきた．VANET が目指す主なセキュリティ要件は，参加ノードを認証することにより，ネットワークの機密性を確保すること，およびルーティングで交換する情報の完全性を保証することである．その上で，これらのセキュリティ要件を満たす機構の追加による通信性能の低下を抑制できることが望ましい．

参加ノードの認証に関しては，Ying と Nayak (2014)が動的なログイン ID とパスワードを利用した認証プロトコルを発表した．参加ノードやルーティング情報の認証を電子署名で行う研究もなされており，それらの研究では標準署名方式である ECDSA が採用されている(Ravi ら(2013))．こうした中央集権的な認証の場合，認証サーバを必要とする．また，SDN (Software Defined Networking)では SDN コントローラーが中央集権的にルーティングを行う方法もあり，Al-Heety ら(2020)は，SDN コントローラーに認証機構を組み込むことにより VANET を形成するという提案をしている．2022 年には，中央集権的な認証を行わない証明書レスの ECDSA も提案されている(Imghoure ら(2022))．いずれの場合も認証機構の追加によりネットワークにおける通信の可用性が確保できるかどうかが課題となっており，ネットワーク形態やルーティング方式ごとに通信性能への影響を調査する必要がある．そこで，本研究では，実世界ノード移動モデルを用いたシミュレーションにより，デジタル署名による認証機構の導入が通信性能に及ぼす影響を明らかにする．

VANET の通信形態の一つである V2V アドホック通信では，位置情報ベースのルーティングプロトコルである GPSR (Greedy Perimeter Stateless Routing)がよく用いられる．本研究では，GPSR のルーティング情報の正当性を保証するため，DSA (Digital Signature Algorithm)と ECDSA (Elliptic Curve Digital Signature Algorithm)の 2 つのデジタル署名を用いて，ノードと位置情報の認証機構を導入した．また 2 つの認証機構の有効性，ネットワークへの負荷，同一エリア内のノード数に対する拡張性を調査するために，離散型ネットワークシミュレータ ns-3 を使用してシミュレーション実験を行った．

本論文は 4 章で構成される．第 1 章では，本研究の対象である VANET や導入するデジタル署名について説明する．第 2 章では，本研究で提案する認証機構の導入方法について述べ，第 3 章でシミュレーションの環境について述べた後，最後に第 4 章でシミュレーション実験の内容と評価項目を示し，実験結果を考察する．

第1章　　VANET とデジタル署名

この章では本研究において重要となる用語の説明をする．1.1 節では研究対象である VANET について，1.2 節では VANET で主に用いられるルーティングプロトコル GPSR について述べる．1.3 節では，本研究で用いた署名アルゴリズム ECDSA の理解に必要となる楕円曲線とその上での演算について述べる．最後に 1.4 節で，デジタル署名とそのアルゴリズムについて詳しく解説する．

1.1　Vehicle Ad Hoc Network

VANET (Vehicle Ad Hoc Network)とは，車両同士の通信で構成されるモバイルアドホックネットワーク[1]である．車両同士の通信(Vehicle-to-Vehicle)，および車両と路側機の通信(Vehicle-to-Infrastructure)で構成され，基地局などの固定のインフラストラクチャに依存しない．移動中の車両を想定しているため，このネットワークはノードの追加や削除が頻繁に発生する動的な環境下で機能する．各車両は原則として全て対等なノードで，エンドノードだけでなく中継ノードとしての役割ももっている．図 1.1 のように，送信元となる車両の電波伝搬範囲外に宛先車両が位置している場合でも，他の車両を経由することでデータを送ることができる．このような通信を**マルチホップ通信**という．

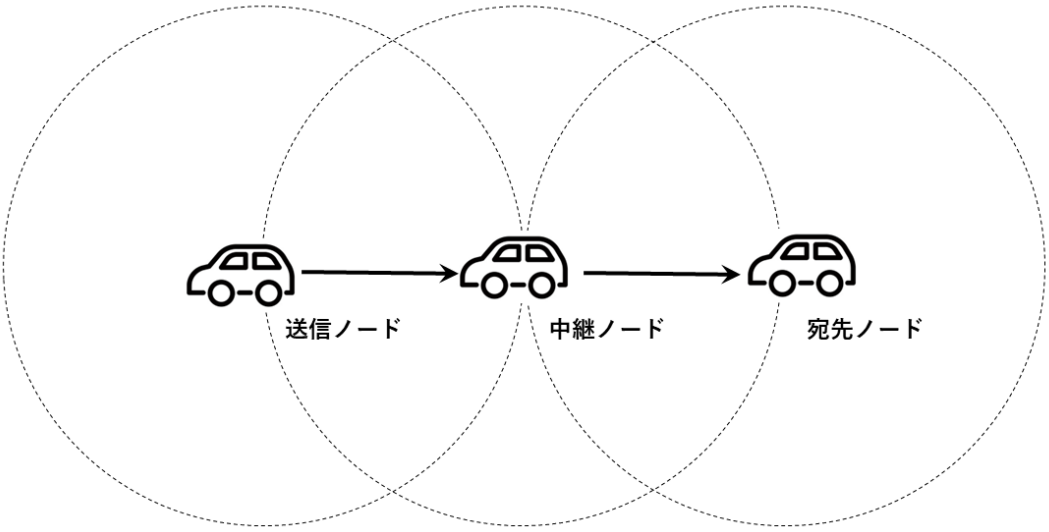

図 1.1　マルチホップ通信

VANET には大きく分けて，次の 4 つの特徴がある．

(1) 急速に変化するネットワークトポロジー

無線通信では，ノード同士が直接的に通信可能であることを**接続している**といい，ノードの接続状態をネットワークで表す．VANET では想定する通信ノードが車両であるため，ノードの移動速度が速く，それに伴ってノード間の接続状態も変化する．そのため，VANET のネットワークトポロジーも急速に変化する．

(2) 十分な電力

車両はエンジンによって継続的に電力を得ることができるため，携帯端末に比べて電力不足の必要がない．しかし，今後動画のやり取りなどの大容量通信の増加や，長距離移動での実現を考慮する場合，電力不足が問題になる可能性がある．

(3) 自身の位置情報取得

車両は，GPS を標準搭載しているため，自身の位置情報を取得することができる．

(4) 移動の制約

車両の動きは道路や建造物によって制限される．

VANET の実用化には高く安定した通信性能を提供すると同時に，通信のプライバシーおよびセキュリティを確保することが不可欠である．過去十年にわたり，セキュリティ確保に対する様々な方法が研究されてきたが[2-8]，いずれの方法においても認証機構の追加による通信の可用性への影響が課題となっている．

1.2 Greedy Perimeter Stateless Routing

GPSR (Greedy Perimeter Stateless Routing) [9]は，VANET で使われている位置情報利用型ルーティングプロトコルである．GPSR では，自身の位置や IP アドレスなどの情報をのせた Hello パケットを一定間隔で隣接ノードに送信する．図 1.2 に示すように，それぞれのノードには IP アドレスや隣接ノードの位置などの情報が記載されたノードテーブルをもち，受信した Hello パケットの情報を用いてノードテーブルを更新することにより隣接ノードの情報を把握する．これらの隣接ノードの位置情報を用いて，Greedy Forwarding と Perimeter Forwarding を組み合わせてルーティングを行う．

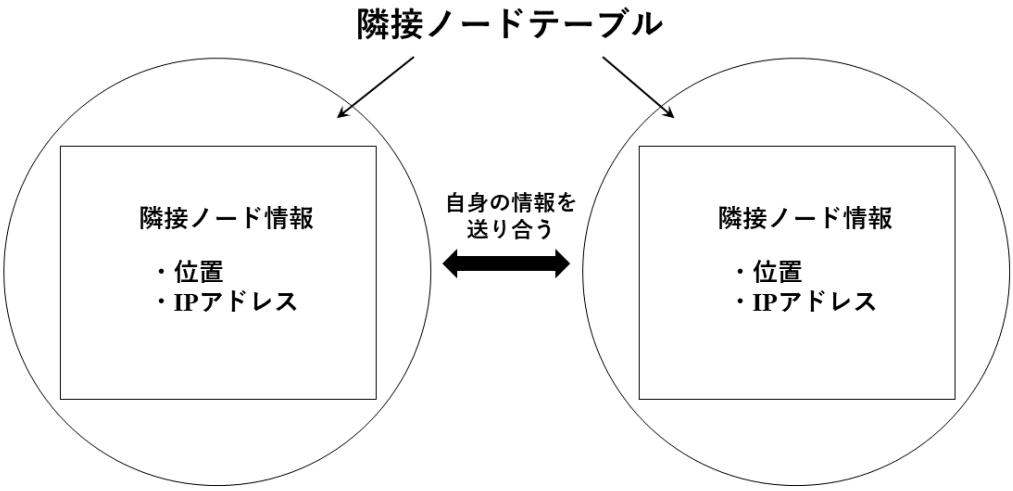

図 1.2　GPSR の隣接ノードテーブル

1.2.1　Greedy Forwarding

GPSR では，基本の転送方法として **Greedy Forwarding** が使用される．図 1.3 を用いて説明する．送信ノード S は，電波伝搬範囲内のノードで宛先ノード D に最も近いノード A を次ホップとして選択する．点線で書かれた円は全ノードの受信感度が等しい場合の送信ノード S の電波伝搬範囲を表している．破線は宛先ノードとの距離を表している．

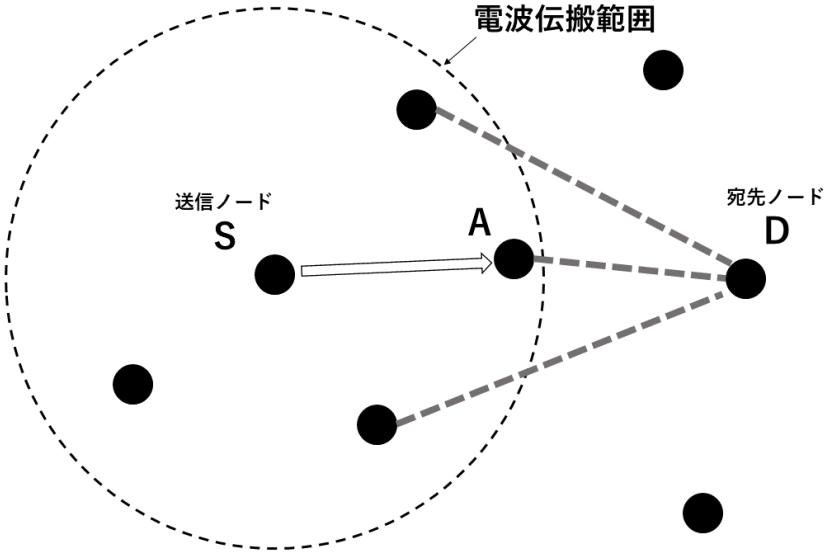

図 1.3　Greedy Forwarding

Greedy Forwarding には**局所最大問題**が存在することが知られている. 局所最大問題とは, 図 1.4 に示すように, 送信ノード S 自身が電波伝搬範囲内で宛先ノード D に一番近い場合, 選択できる次ホップが存在しなくなる問題である.

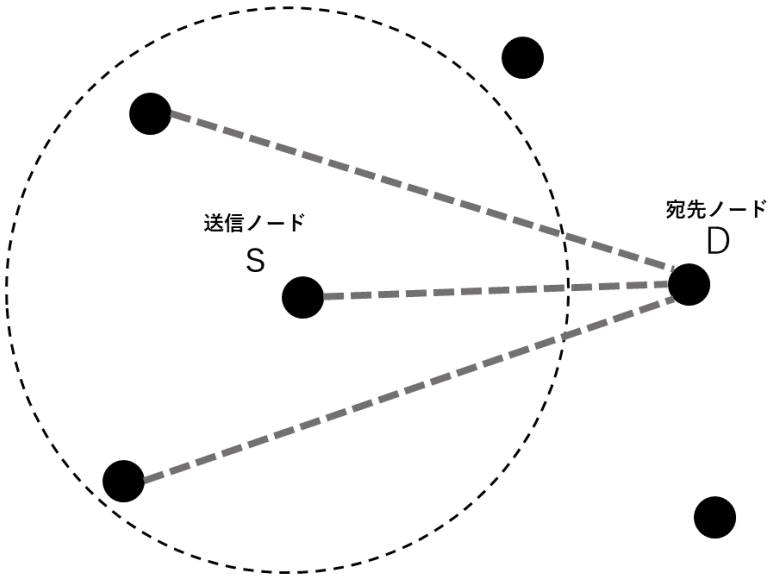

図 1.4　局所最大問題

1.2.2　Perimeter Forwarding

Greedy Forwarding で局所最大問題が生じるような場合には，**Perimeter Forwarding** が使用される．図 1.5 に示すように，送信ノード S を中心に **Right Hand Rule** に則って反時計回りにノードを探索し，最初に見つけたノード A を次ホップとして選択する.

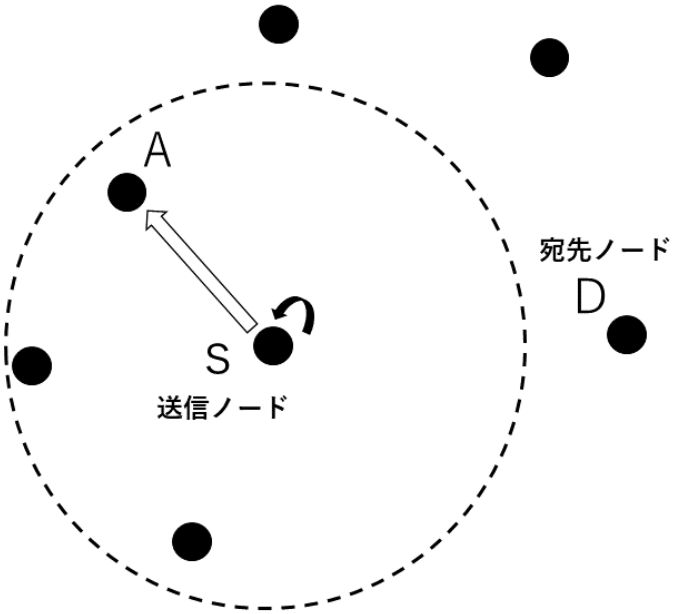

図 1.5　Perimeter Forwarding

1.3　楕円曲線

体$\mathbb{F}_p$ (pは素数)上で定義された**楕円曲線**とは，$a_1, a_2, \cdots, a_6 \in \mathbb{F}_p$に対し，以下のように定義される平面 3 次曲線のことである．

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

特に，$p \neq 2, 3$のとき，標準形

$$y^2 = x^3 + ax + b \ (a, b \in \mathbb{F}_p) \tag{1.1}$$

に変形できることが知られている．式(1.1)が特異点(尖点や自己交差点)をもたないためには，以下の判別式Δが 0 であってはならない．なお，楕円曲線が非特異であることは，この後で紹介する楕円加算が矛盾なく定義できるために必要な性質である．

$$\Delta = -16(4a^3 + 27b^2) \neq 0.$$

楕円曲線上の点のうち，x, y座標がともに有限体$\mathbb{F}_p$の元となる点(x, y)を**有理点**という．また，楕円曲線には**無限遠点**と呼ばれる特殊な点$\mathcal{O}$が存在を仮定し，これも楕円曲線上の有理点の集合に含める．楕円曲線上の有理点の集合は，以下のような演算法則に関して群をなす．楕円曲線暗号方式(署名方式を含む)では，この性質が利用される．

・楕円加算

楕円曲線E上の任意の 2 点$P(x_1, y_1)$, $Q(x_2, y_2)$の加算は以下のように行う．

(i) 点P, Qを通る直線がy軸と並行でないとき：

Step 1.　点P, Qを通る直線Lを引く．

$$L: y - y_1 = \frac{y_1 - y_2}{x_1 - x_2}(x - x_1).$$

Step 2.　楕円曲線Eと直線Lは，点P, Qの他にもう 1 つ交点をもつ．この交点を$R(x_3, y_3)$とする．

$$x_3 = \left(\frac{y_1 - y_2}{x_1 - x_2}\right)^2 - x_1 - x_2,$$
$$y_3 = \frac{y_1 - y_2}{x_1 - x_2}(x_1 - x_3) - y_1.$$

Step 3.　点Rをx軸対称な点$R'(x_3, -y_3)$が$P + Q$となる．

$$P + Q = R'.$$

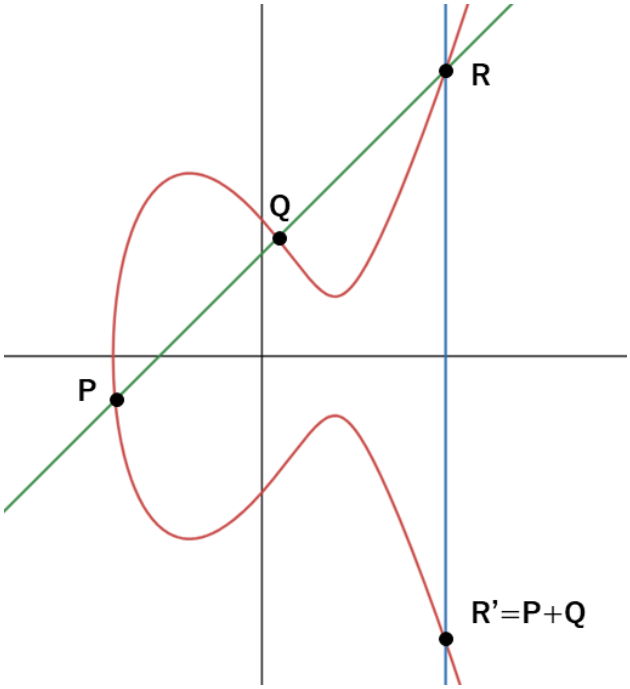

図 1.6　楕円曲線上の異なる 2 点の加算

(ii) 点P, Qを通る直線がy軸と並行であるとき：

無限遠点$\mathcal{O}$が$P + Q$となる.

$$P + Q = \mathcal{O}.$$

(iii) $P = Q$のとき：

Step 1.　点Pにおける接線L'を引く.

$$L' : y - y_1 = \frac{3x_1^2 + a}{2y_1}(x - x_1).$$

Step 2.　楕円曲線Eと直線L'の交点を$R(x_3, y_3)$とする.

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1,$$
$$y_3 = \frac{3x_1^2 + a}{2y_1}(x_1 - x_3) - y_1.$$

Step 3.　点Rをx軸対称な点$R'(x_3, -y_3)$が$P + Q$となる.

$$P + Q = 2P = R'.$$

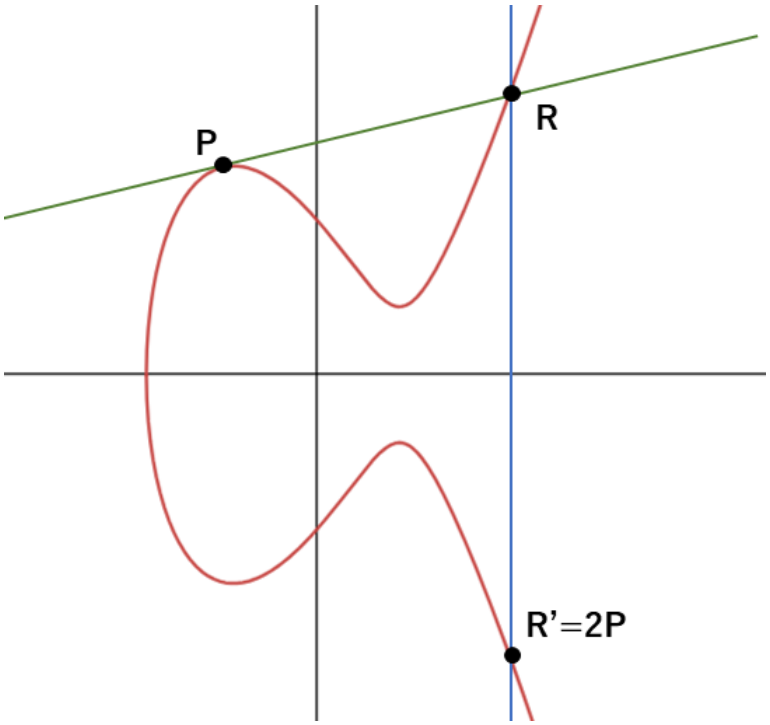

図 1.7　楕円曲線上の同一点の加算

(iv) $P = \mathcal{O}$または$Q = \mathcal{O}$のとき：

無限遠点は加算において加法単位元の役割を果たす.

$$\mathcal{O} + Q = Q,$$
$$P + \mathcal{O} = P.$$

・スカラー倍算

正整数kによる点Gのスカラー倍$P = kG$は，点Gをk回加算することで求められる.

$$P = kG = \underbrace{G + \cdots + G}_{k\text{ 回}}.$$

また，$kG = \mathcal{O}$ となる最小の正整数kをGの**位数**という.

ある基準点Gに対し，$P = kG$となる楕円曲線上の点Pが与えられたとき，kとGからPを求めるのは容易だが，GとPからkを求めるのは困難であることが知られている．これを楕円曲線上の**離散対数問題**という.

1.4　デジタル署名

デジタル署名は，電子文書やメッセージの真正性と完全性を検証するために使われる暗号技術である．デジタル署名を利用することにより，メッセージ作成者の本人証明や，受信したメッセージの非改ざん性(送信時と受信時のメッセージが一致していること)が保証される．

一般に，デジタル署名方式は3つのフェーズに分けられる．

1.　鍵生成フェーズ

署名者(送信者)により，署名のための 1 対の鍵，検証鍵(公開鍵)と署名鍵(秘密鍵)，が生成される．署名鍵は署名者のもとで秘密に保管され，検証鍵は検証者(受信者)に公開される．

2.　署名生成フェーズ

メッセージ作成者(送信者)は，メッセージからハッシュ値を計算する．ハッシュ値は元のメッセージが少しでも変更されると全く異なる値となるため，完全性の確認に用いられる．メッセージ作成者(送信者)は，自身の署名鍵(秘密鍵)を使用してハッシュ値に署名する．メッセージ作成者(送信者)以外が署名鍵を求めることは計算量的に困難なため，署名はメッセージ作成者(送信者)の真正性の証明となる．

3.　署名検証フェーズ

検証者(受信者)は検証鍵(公開鍵)を使用して署名を検証する．検証鍵は署名者の署名鍵と対になるものであるため，この検証プロセスを通じて，メッセージがメッセージ作成者(送信者)により署名され，送信時点から改ざんされていないことが確認できる．

デジタル署名のアルゴリズムには様々な種類がある．次節では，本研究で使用した米国連邦情報処理標準である DSA と ECDSA について詳しく説明する．

1.4.1　DSA

DSA (Digital Signature Algorithm)は, 1993 年に National Institute of Standards and Technology (NIST)により FIPS 186 として標準化された主要な電子署名方式の 1 つである．なお，FIPS 186-5 [10]では，DSA は新たにデジタル署名を行うことには推奨されないが，標準策定以前に行われた署名の検証には引き続き利用可能とされている．

DSA の 3 つのフェーズにおける処理は以下の通りである．

鍵生成フェーズ

Step 1. セキュリティパラメータとして，L, N $(L > N)$を定める．
FIPS 186-4 [11]では，以下の 4 つの組が規定されている．
$(L, N) = (1024, 160), (2048, 224), (2048, 256), (3072, 256)$.

Step 2. N bit の素数q $(2^{N-1} < q < 2^N)$，L bit の素数p $(2^{L-1} < p < 2^L)$をランダムに選ぶ．ただし，$q \mid (p-1)$を満たすものとする．

Step 3. ランダムな整数h $(2 \leq h < p)$に対して，
$$g \equiv h^{(p-1)/q} \pmod{p}$$
とする．$g = 1$となる場合には，hを選択し直し，Step 3 を再度実行する．

Step 4. ランダムな整数x $(2 \leq x < q)$に対して，
$$y \equiv g^x \pmod{p}$$
を計算する．

Step 5. p, q, gを専用のパラメータとして，yを検証鍵として公開する．xは署名鍵として安全に管理する．

Step 3 では，素数位数pの有限体$\mathbb{F}_p$において，位数がqとなる元を探索している．qを用いず(したがって，Step 3 を行わず)，gを$\mathbb{F}_p$の原始元として，Step 4 で$2 \leq x \leq p-2$を満たす乱数xに対し，$y \equiv g^x \pmod{p}$とすることもできるが，有限体$\mathbb{F}_p$における位数qの部分群を利用することで，安全性を損なうことなく，計算量をLビットからNビットに削減することができる．

Step 4 においてxを 2 以上とする理由は，$x = 1$の場合，検証鍵から署名鍵が直ちに明らかとなるからである．

署名生成，検証には，少なくともNbit の出力をもつ暗号学的ハッシュ関数が必要となる．任意長のメッセージMに対する署名は，パラメータp, q, g，検証鍵x，あらかじめ検証者との間で合意したハッシュ関数Hを用いて次のように生成する．

署名生成フェーズ

Step 1. $2 \leq k \leq q-2$ を満たすランダムな整数kを選ぶ.

Step 2. $r \equiv \left(g^k \ (mod\ p)\right)\ (mod\ q)$を計算する.

Step 3. $s \equiv\ k^{-1}\ (H(M) + xr)\ (mod\ q)$を計算する.

Step 4. $r = 0$または$s = 0$の場合，Step 1 まで戻りkを選び直し，再計算を行う.

Step 5. Step 3, 4 で計算した(r, s)のペアをメッセージMに対する署名とし，Mとともに受信者に送信する.

メッセージMに対する署名(r, s)を検証するには，パラメータp, q, g，検証鍵，ハッシュ関数Hを用いて，以下のアルゴリズムを実行する.

署名検証フェーズ

Step 1. メッセージMと署名の組(r, s)を受信し，$0 < r < q$かつ$0 < s < q$であるかを調べる．これらを満たしていない署名は棄却する.

Step 2. $w \equiv\ s^{-1}\ (mod\ q)$を計算する.

Step 3. $u_1\ \equiv wH(M)\ (mod\ q)$を計算する.

Step 4. $u_2\ \equiv rw\ (mod\ q)$を計算する.

Step 5. $v\ \equiv \left(g^{u_1} y^{u_2}\ (mod\ p)\right)\ (mod\ q)$を計算する.

Step 6. $r\ \equiv v\ (mod\ q)$ならば署名を受理する．そうでなければ署名は不正とみなし，棄却する.

正当なメッセージと署名の組$(M, (r, s))$に対して，

$$g^{u_1} y^{u_2} =\ g^{u_1 + x u_2} =\ g^{(H(M) + xr)w} \equiv\ g^k\ (mod\ p)$$

が成り立つ．これは，

$$\begin{aligned} v &\equiv \left(g^{u_1} y^{u_2}\ (mod\ p)\right)\ (mod\ q) \\ &\equiv \left(g^k\ (mod\ p)\right)\ (mod\ q) \\ &\equiv r\ (mod\ q) \end{aligned}$$

であることを示している．これにより，DSA 署名が正しく機能することがわかる．ただし，アルゴリズムが正しく機能することと，安全であることは別である.

攻撃者が$r \equiv v\ (mod\ q)$を満たすようにメッセージMやその署名(r, s)を都合よく改ざんできれば，その攻撃は成功する．しかし，rには署名者しか知らない秘密の乱数kが離散対数として使用されており(署名生成フェーズの Step 2)，十分なビット長をもつp, qのもとでは，これらの値の改ざんは計算量的にほぼ不可能と考えられる．

1.4.2 ECDSA

ECDSA (Elliptic Digital Signature Algorithm)は，楕円曲線上の演算を用いて，DSA を実現する署名方式である．ECDSA は，処理速度と安全性において有限体上の DSA よりも優位性がある．短い鍵長で同等の安全性を確保できるため，計算量が少なく，低性能なデバイスでも高速な処理が可能であり，かつ演算規則の複雑さから攻撃者が鍵を推測することも困難であることがその理由である．

ECDSA の 3 つのフェーズにおける処理は以下の通りである．

鍵生成フェーズ

Step 1. 法とする素数pと楕円曲線Eを選択し，E上の基準点Gを選ぶ．

Step 2. 検証鍵dを$2 \leq d \leq n-1$の範囲からランダムに選ぶ．ただし，nはGの位数である．

Step 3. 署名鍵を$Q = dG$と計算する．

署名生成フェーズ

Step 1. kを$2 \leq k \leq n-1$の範囲からランダムに選び，kGのx座標をrとする．

Step 2. メッセージMのハッシュ値$h = H(M)$を計算する．ここで，Hは検証者と事前に合意しているハッシュ関数である．

Step 3. 以下のようにsを求め，Mに対する署名を(r, s)として，検証者にMとともに送信する．

$$s \equiv k^{-1}(h + dr)\ (mod\ n).$$

署名検証フェーズ

Step 1. 検証者は，メッセージMとMに対する署名(r, s)を取得し，ハッシュ値$h = H(M)$を算出した後，

$$u \equiv s^{-1}h \ (mod\ n), \qquad v \equiv s^{-1}r \ (mod\ n)$$

とする．

Step 2. 楕円曲線上の点として，

$$Q' = (x', y') = uG + vD$$

を求め，Step 1 で計算したQ'のx座標がrと一致した場合，署名を受理する．そうでなければ不正な署名とみなし，署名を棄却する．

ECDSA が正しく機能することは，以下のようにして確かめられる．

$$\begin{aligned} Q' &= uG + vQ \\ &= hwG + rwQ \\ &= hwG + rwdG \\ &= hwG + (sw - h)G \\ &= swG \\ &= kG. \end{aligned}$$

以上より，$r = x'$が成立する．安全性に関する考え方は，DSA 署名とほぼ同様であるが，ECDSA の安全性は，楕円曲線上の演算の複雑性からも強化されると考えられる．

第2章　　提案手法

位置情報利用型ルーティングプロトコルである GPSR で正しいルーティングが行われる条件は，同一アドホックネットワークの参加者が相互に正しい位置情報を送信し合うことである．そのため，以下の 2 種類の不正ノードによる攻撃が想定される．

1.　位置情報詐称ノード

位置情報詐称ノードとは，ルーティングを妨害し，通信データを窃取することを目的とした内部不正ノードである．**内部不正ノード**とは，悪意を持ったネットワーク参加者のことである．図 2.1 では，送信ノード S が宛先ノード D に送信しようとしている．このとき，S は D に最も近い電波伝搬範囲内のノードであるノード B を次ホップに選択するべきである．しかし，A が本来の位置情報ではなく，B よりも D に近い位置情報を詐称して S に送った場合，S は A の位置を誤って把握し，A にパケットを送信してしまう．このようにして A は本来 D が受け取るべき情報を窃取できる．つまり，A は位置情報を詐称することにより，D への情報伝達を妨害できることになる．

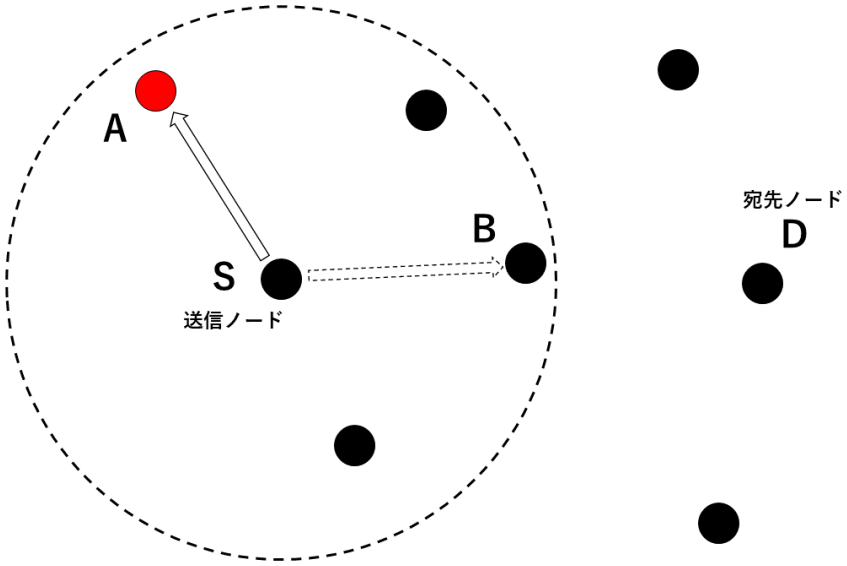

図 2.1　　位置情報詐称ノード

2. IP アドレス詐称ノード

IP アドレス詐称ノードは，ネットワークへの不正アクセスを目的とした外部不正ノードである．**外部不正ノード**とは，ネットワークに参加していない悪意をもったノードのことである．図 2.2 では，当該ネットワークの構成ノードを黒丸で，非構成ノードを赤丸で示している．赤丸で表した外部ノードがネットワーク内の IP アドレスを詐称してネットワーク参加者に送信した場合，その情報を受信したネットワーク参加者は外部ノードをネットワーク参加者と誤って把握し，通信を始めてしまう．このように，IP アドレスを詐称することで外部ノードがネットワークに不正に参加できる可能性がある．

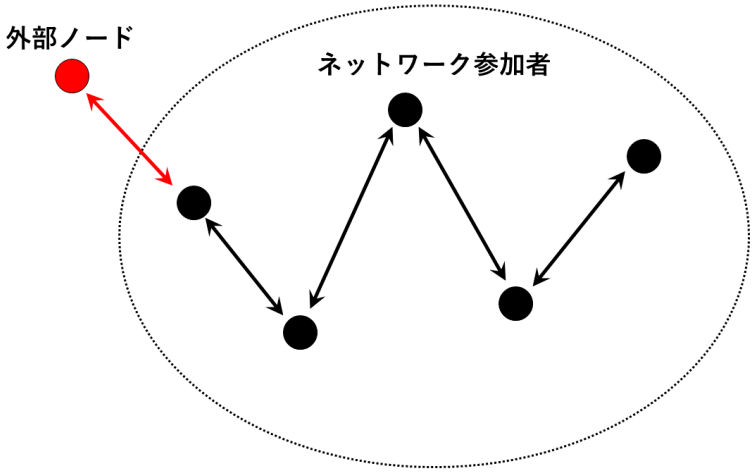

図 2.2　IP アドレス詐称ノード

このような 2 種類の不正ノードが存在する場合，正しくルーティングが行われない．そこで，本研究では，デジタル署名を用いてノード情報の検証を行い，なりすましや改ざんを排除するようにした[12, 13]．なお，一般にはメッセージ作成者(送信者)が署名者でもあるのに対し，本研究で提案する機構では，署名の送信者と署名者が異なる．

その仕組みを，具体的に図 2.3 を用いて説明する．データがノード A からノード B に送信される場合，A は認証局の署名付き証明書をデータに付与して B に送信する．受信側の B は，受信したデータが

・A 本人から送られてきたか
・改ざんされていないか

を署名を検証することで確認する．

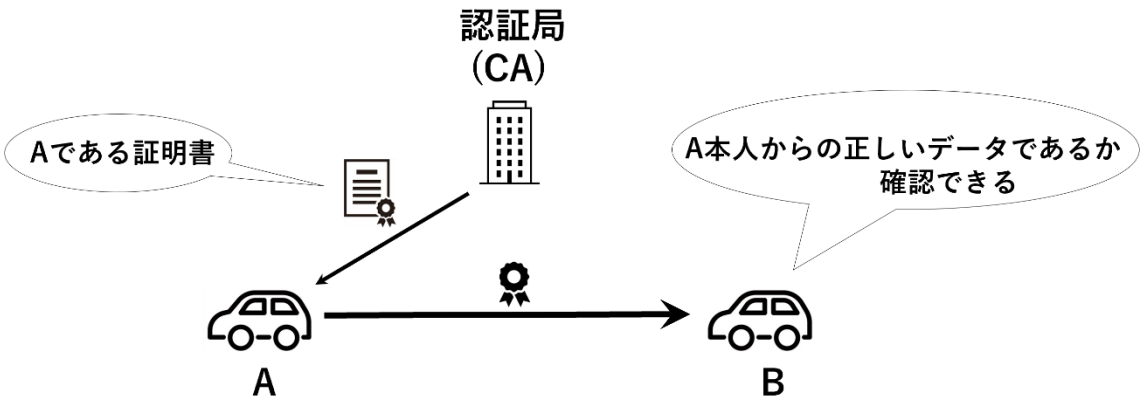

図 2.3　本研究でのデジタル署名の使用方法

本研究では，位置情報と IP アドレスを認証する認証機構を導入する．導入方法は以下の通りである(図 2.4 参照).

デジタル署名を用いた認証機構

Step 1.　各ノードが GPS から位置情報を取得する際，その位置情報が正しいものであると証明できる機関から署名をもらう．

Step 2.　各ノードが DHCP サーバから IP アドレスを取得する際，DHCP サーバから署名をもらう．

Step 3.　隣接ノードと Hello パケットを交換する際に，位置情報と IP アドレスの署名も同時に送信する．

Step 4.　Hello パケットを受信したノードは，2 つの署名の検証を行う．どちらの検証も成功した場合のみ隣接ノードテーブルを更新する．どちらか 1 つでも検証に失敗した場合，受け取った Hello パケットを破棄する．

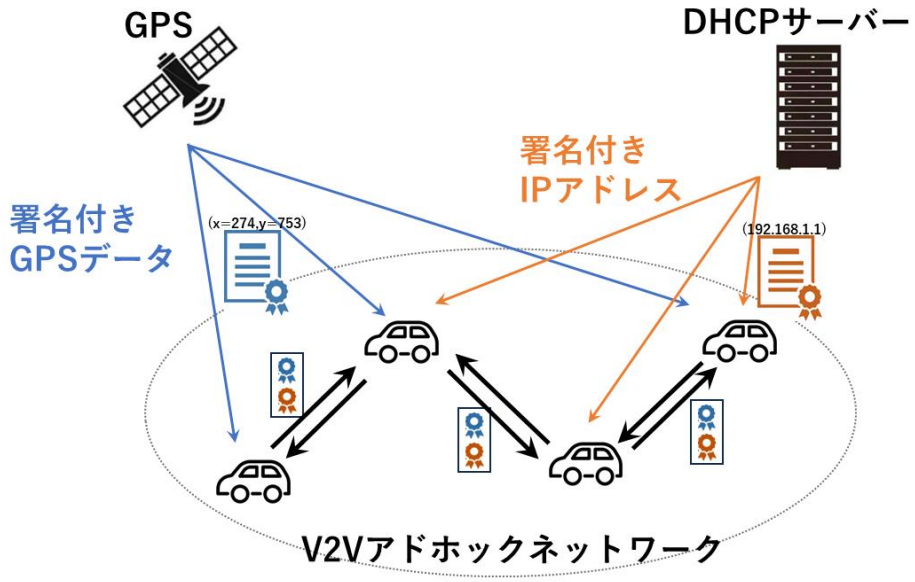

図 2.4　デジタル署名を用いた認証機構

認証局(署名者)と被署名者データの対応を表 2.1 に示す.

表 2.1　認証局(署名者)と被署名データの対応

認証局(署名者)	被署名データ
DHCP サーバ	IP アドレス
位置情報が正しいものと証明できる機関	位置情報

第3章　　シミュレーション環境

この章では，本研究で行ったシミュレーション環境について述べる．シミュレーションパラメータは表 3.1 に示す通りである．

表 3.1　シミュレーションパラメータ

シミュレーションツール	ns-3.26
通信プロトコル	UDP
通信規格	IEEE802.11p
パケットサイズ	1024[bytes]
送信電力	17.026[dBm]
パケット送信間隔	1.0[s]
電力検出閾値	-96[dBm]
電波伝搬減衰モデル	対数距離電波伝搬減衰モデル
遅延モデル	ConstantSpeedPropagationDelayModel
ノード数(個)	40 (実験 1, 2)，37/74/112/148/185 (実験 3)
電波伝搬範囲	約 300[m]
シミュレーション時間	300[s]
ルーティングプロトコル	GPSR
デジタル署名	DSA，ECDSA

3.1　ネットワークシミュレータ ns-3

本研究で行ったシミュレーションには，シミュレーションツールとして **network simulator-3 (ns-3)**[14]を使用した．ns-3 は，オープンソースの通信シミュレータソフトウェアである．有線，無線両方を含む様々なプロトコルをシミュレートできる能力をもっており，Wi-Fi，LTE，TCP/IP などが含まれる．実環境との統合も可能であり，シミュレータと実機間でパケットを送受信することもできる．開発言語は C++と Python が用いられているが，コアファイル，シナリオファイルは共に C++で書かれている．本研究ではモデルを利用するため，C++でコーディングした．

ns-3 では，様々なコンテナを組み合わせてプログラムが構築される．コンテナによって管理されたものをグループ化し，操作を容易にする．一般的に使用される主要なコンテナを以下に示す．

・**NodeContainer**

ノードを管理するためのコンテナである．ns-3 のノードとは，コンピュータ，ルータなどシミュレートされるネットワーク内の個々のデバイスを表す.

・**DeviceContainer**

ネットワークインターフェースカードやその他の通信デバイスを管理するためのコンテナである.

・**InterfaceContainer**

IP インターフェースを管理するためのコンテナである.

・**ApplicationContainer**

アプリケーションレイヤのプログラムやサービスを管理するためのコンテナである．ns-3 には，HTTP サーバや FTP クライアントなどの様々な種類のアプリケーションが用意されている.

ns-3 は Linux 環境での動作を前提に開発されている．そのため，本研究では仮想環境 VMware に ubuntu をインストールし，その上で ns-3 を動作させた．現時点での ns-3 の最新バージョンは 20 23 年 9 月 27 日リリースの ns-3.40 となっている．本研究では，先行研究[15]と互換性のある ns-3.26 を使用した.

3.2 通信規格 IEEE802.11p

本研究では，通信規格として IEEE802.11p [16]を使用した．**IEEE802.11p** とは，Intelligent Transportation System (ITS)での通信に特化した通信規格 IEEE802.11a ベースの無線 LAN 規格である．この規格の特徴を以下に示す.

・**周波数**

この規格は 5.9GHz 帯の周波数帯域で動作する．この帯域は ITS アプリケーション専用に割り当てられているため，他の無線デバイスとの干渉を最小限に抑えることができる．また，10MHz のチャネル幅を利用した 7 つのチャネルをもつ.

・**通信速度**

通信速度は周囲の環境やチャネルの状態によって 3Mbps から 27Mbps の範囲となる.

・アクセス方式

アクセス方式に CSMA/CA を利用している．**CSMA/CA** は，複数のデバイスが同じ通信チャネルを共有する環境で，データパケットの衝突を回避するために設計されている．

・変調方式

変調方式は OFDM を利用している．OFDM は，高速で信頼性の高いデータ伝送を可能にするデジタル信号の変調方式の一つである．

3.3 対数距離電波伝搬減衰モデル

ns-3 では，電波の伝搬損失を計算するために様々な電波減衰モデルがある．本研究では，都市，郊外，屋内といった様々な環境に適用できる**対数距離電波伝搬減衰モデル**を使用した．このモデルは ns-3 上で ns3::LogDistancePropagationLossModel と表現される．

対数減衰モデルの定義を式(3.1)に示す．ここで，dは送信機と受信機間の実際の距離，d_0は参照距離，Lは距離dでの伝搬損失(dB)，L_0は参照距離d_0での伝搬損失，nは環境依存のパケットロス指数である．

$$L = L_0 + 10nlog_{10}\left(\frac{d}{d_0}\right). \tag{3.1}$$

3.4 ConstantSpeedPropagationDelayModel

本研究では遅延モデルとして，電波伝搬における物理的な過程をシンプルに再現している ConstantSpeedPropagationDelayModel を使用した．このモデルは電波が空間を伝搬する速度が一定であるという仮定に基づいている．伝搬遅延Δtは，送信機と受信機の距離dと伝搬速度vを用いて式(3.2)で定義される．

$$\Delta t = \frac{d}{v}. \tag{3.2}$$

3.5　移動モデル

本研究では，シミュレーション環境を実世界により近づけるために，**Simulation of Urban Mobility(SUMO)** [17]と **OpenStreetMap** [18]を利用して名古屋駅近辺の交通データを取得し，実際の車両の動きを再現するノードのモビリティモデルを作成した．

SUMO は都市環境における交通流動をシミュレーションするためのオープンソースソフトウェアである．与えられた交通ネットワークから自動車，バス，電車などで構成されている交通流をシミュレーションすることができる．

移動モデルの作成の流れを以下に示す．

Step 1.　OpenStreetMap のウェブサイトから，シミュレーションに使用したい地域の地理データを取得する．図 3.1 は OpenStreetMap で表示した名古屋駅近辺である．Select Area で枠を区切り，duration でシミュレーション時間を決定し，Generate Scenario ボタン押すと交通流データが取得できる．

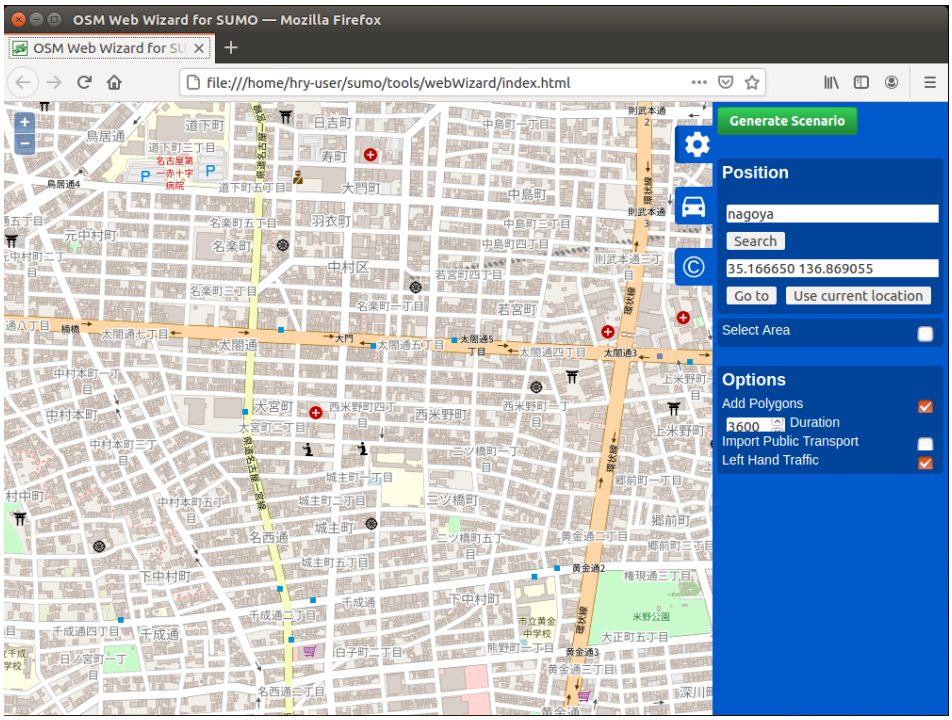

図 3.1　OpenStreetMap で表示した名古屋駅近辺

Step 2.　OpenStreetMap で取得した地理データを, SUMO の読み取れる.xml 形式に変換する．このデータを用いて SUMO はシミュレーションを行う．図 3.2 は.xml ファイルに変換した地理データを SUMO で表示したものである．

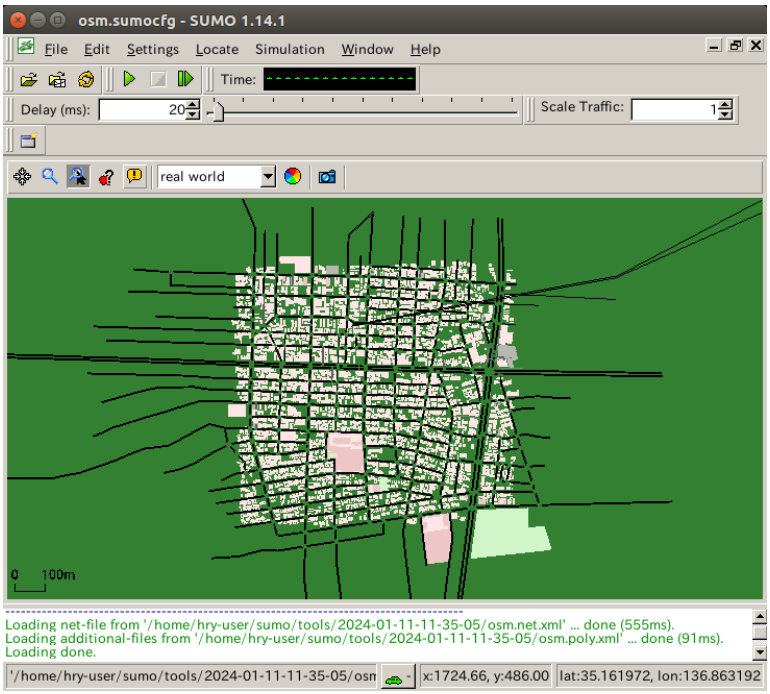

図 3.2　SUMO で表示した名古屋駅近辺

Step 3.　SUMO で行ったシミュレーション，すなわちノードの動きのデータを ns-3 で読み取れる.tcl 形式に変換する．図 3.3 は.tcl 形式に変換したノードの動きのデータを ns-3 のビジュアライザーで表示したものである．

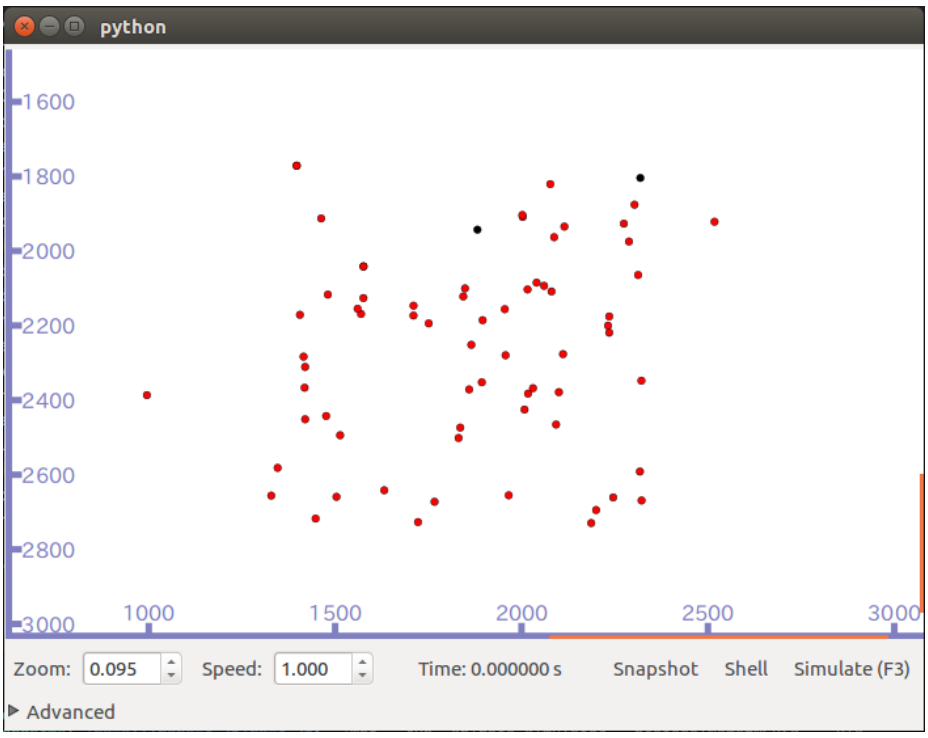

図 3.3　ns-3 のビジュアライザーで表示した名古屋駅近辺

3.6　デジタル署名

本研究ではデジタル署名に，1.4 節で解説した DSA と ECDSA を使用した．DSA のセキュリティパラメータは以下の通りである．

$$(L, N) = (2048, 256).$$

ECDSA で使用した楕円曲線は scep256k1 である．Scep256k1 の式は以下の通りである．

$$y^2 = x^3 + 7.$$

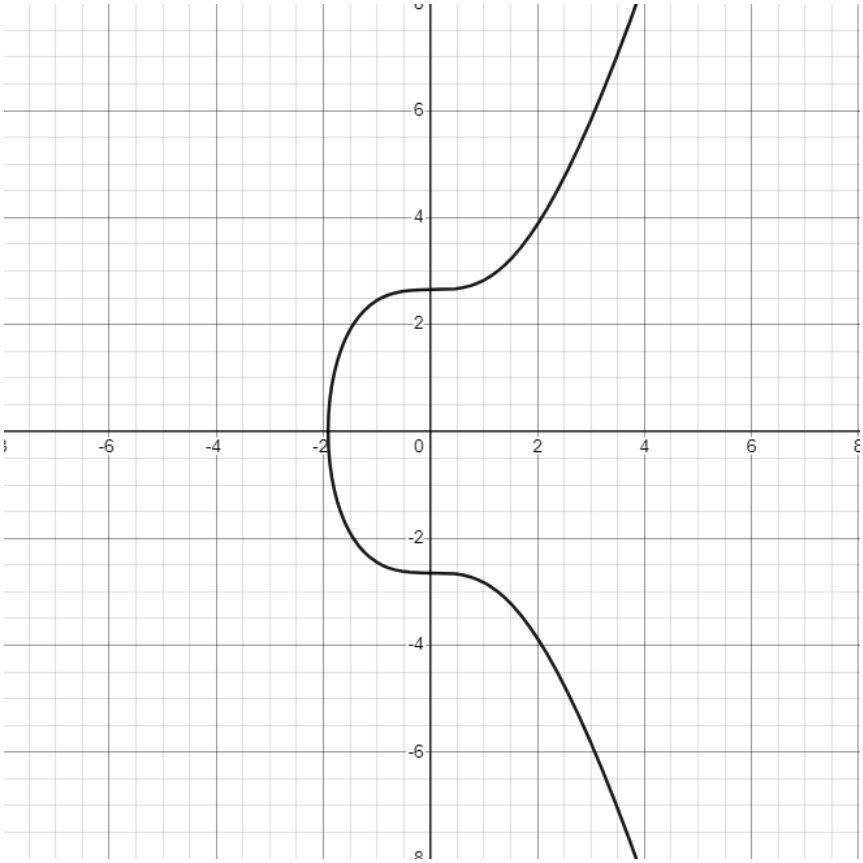

図 3.4　scep256k1 の楕円曲線

第4章　　シミュレーション実験

本研究では，第 3 章で述べたシミュレーション環境を用いて 3 種類のシミュレーション実験を行った．いずれのシミュレーション実験でも，以下の 3 パターンを調べた．

(a) 認証機構を用いない場合
(b) DSA を用いて認証機構を追加した場合
(c) ECDSA を用いて認証機構を追加した場合

本章では，これらのシミュレーション実験の概要とその結果について述べる．シミュレーション結果を評価するのに使用した項目は以下の 3 つである．

1. 平均パケット配送率(PDR)

平均パケット配送率とは，送信データが宛先に到着した割合のことである．定義を式(4.1)に示す．ここで，$TxBytes$は送信ノードの合計送信バイト数，$RxBytes$は宛先ノードの合計受信バイト数である．

$$PDR = \frac{RxBytes}{TxBytes} \times 100[\%]. \tag{4.1}$$

2. スループット(TP)

スループットとは，単位時間当たりに正常に送信されるデータ量のことである．定義を式(4.2)に示す．ここで，$AllTxBytes$は合計送信バイト数，$TxTmes$は送信にかかった時間である．

$$TP = \frac{AllTxBytes}{TxTimes}[kbps]. \tag{4.2}$$

3. オーバーヘッドサイズ(OH)

オーバーヘッドサイズとは，ルーティングに使用される通信データ量のことである．定義を式(4.3)に示す．ここで，$AllTxKBytes$は合計送信キロバイト数，$TxKBytes$はデータパケットの合計送信キロバイト数である．

$$OH = AllTxkBytes - TxkBytes[KB]. \tag{4.3}$$

4.1　実験 1

実験 1 では，認証機構が正しく機能することを確認するための実験を行った．全ノードの 10%を不正ノードに変更して 1000 回シミュレーションを行い，平均パケット配送率とスループットを調べた．導入した不正ノードは第 2 章で述べた位置情報詐称ノードと IP アドレス詐称ノードの 2 種類である．

実験の結果は以下の通りである．

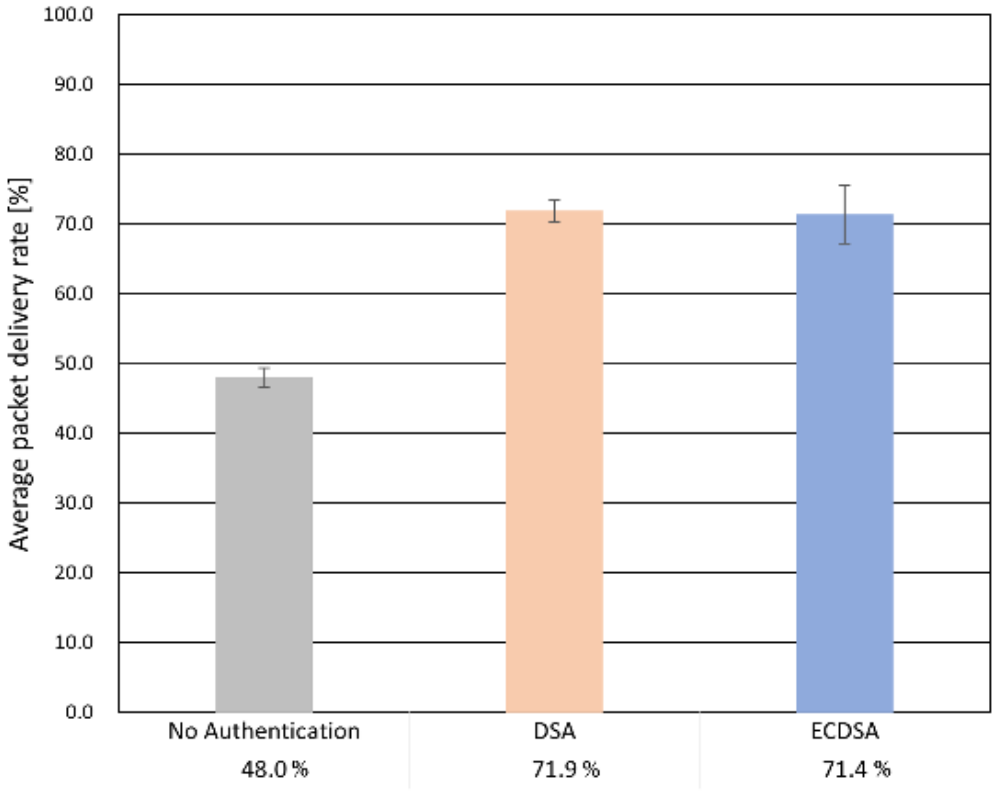

図 4.1　不正ノードが存在する環境での平均パケット配送率

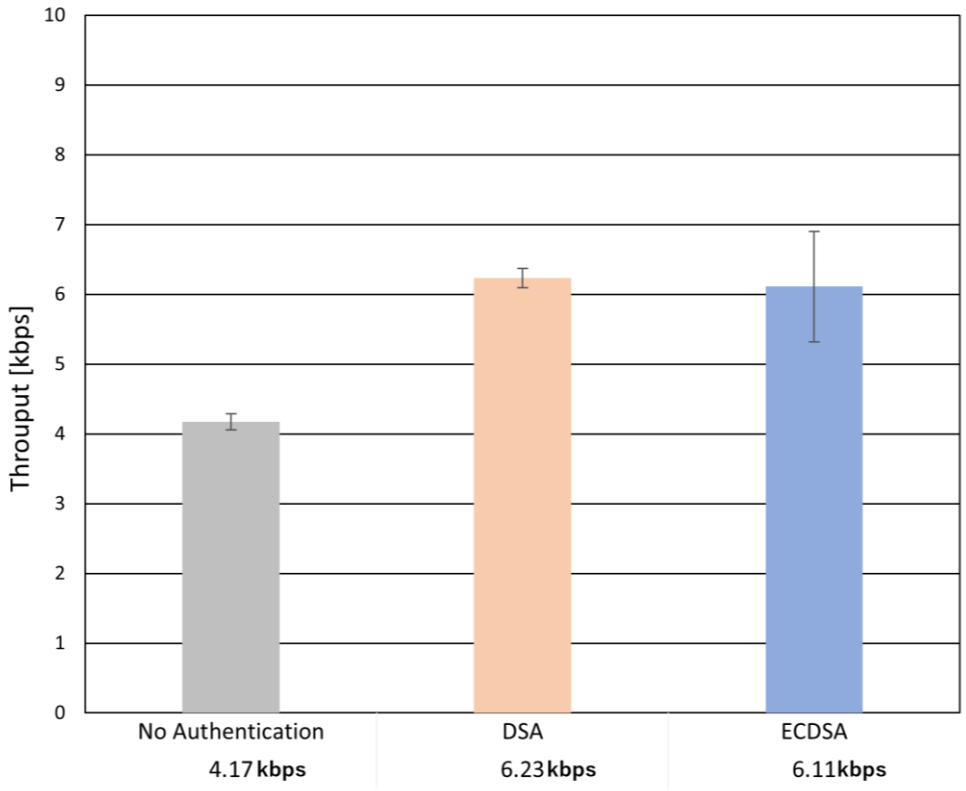

図 4.2　不正ノードが存在する環境でのスループット

図 4.1 は実験 1 におけるシミュレーションパターンごとの平均パケット配送率を示している. 認証機構なしの場合に 48.0%だった平均パケット配送率が, DSA のとき 71.9%, ECDSA のとき 71.4%と，20%以上向上した.

図 4.2 はシミュレーションパターンごとのスループットを示している. 認証機構なしのとき 4.17kbps だったスループットは，DSA のとき 6.23kbps，ECDSA のとき 6.11kbps となっており，平均配送率と同様にスループットも向上したことがわかる.

認証機構を追加した場合の平均パケット配送率とスループットの向上は，認証機構により不正ノードが排除されたことで，ルーティング情報の攪乱による遅延を回避できたためと考えられる．また，DSA，ECDSA のどちらの場合も不正ノードを排除できており，2 つの署名方式が同等のセキュリティ性能をもつことを示唆している.

4.2　実験 2

実験 2 では, 認証機構の追加によるネットワークへの負荷を調べるための実験を行った. 不正ノードの存在しない環境で 1000 回シミュレーションを行い，平均配送率と平均オーバーヘッドサイズを調べた.

実験の結果は以下の通りである.

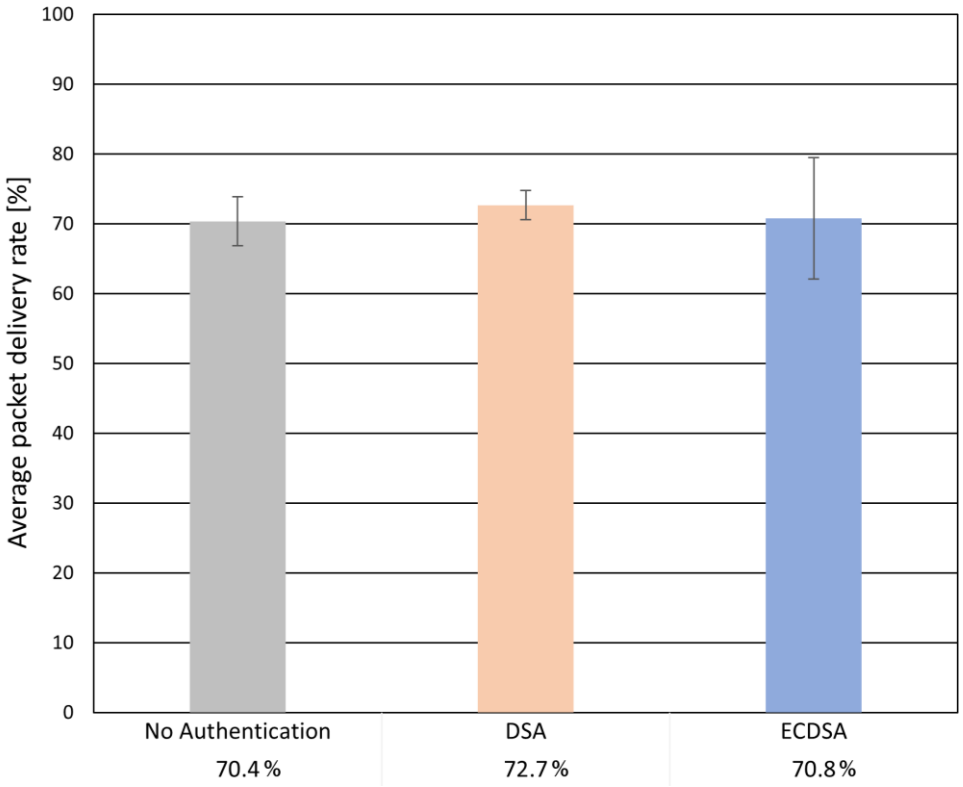

図 4.3　平均パケット配送率

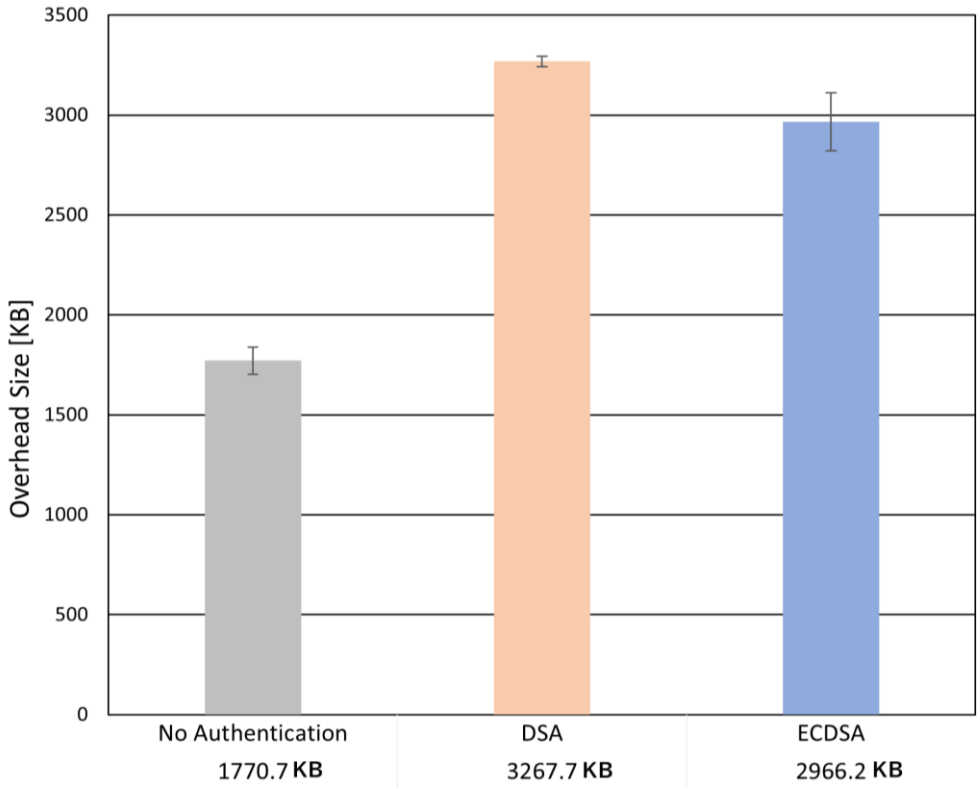

図 4.4　平均オーバーヘッドサイズ

図 4.3 は実験 2 におけるシミュレーションパターンごとの平均パケット配送率を示している．認証なしのとき 70.4%，DSA のとき 72.7%，ECDSA のとき 70.8%であり，不正ノードが存在しない場合，認証機構の有無は配送率に影響を与えていないことがわかる．

図 4.4 はシミュレーションパターンごとの平均オーバーヘッドサイズを示している．認証なしのとき 1770.7KB に対し，DSA のとき 3267.7KB，ECDSA のとき 2966.2KB であり，認証機構の追加によってオーバーヘッドサイズが大幅に増加していることがわかる．これは，Hello パケットのデータに署名が付与されているためと考えられる．ECDSA 認証の方が DSA 認証と比較してオーバーヘッドサイズが小さいのは，各ノードがもつ鍵の長さが ECDSA の方が短いためである．

4.3　実験 3

実験 3 では，認証機構の効率を評価するための実験を行った．不正ノードの存在しない環境でノード数を 37，74，112，148，185 と変化させて 250 回ずつシミュレーションを行い，それぞれの実行にかかった時間を調べた．

実験結果は以下の通りである．

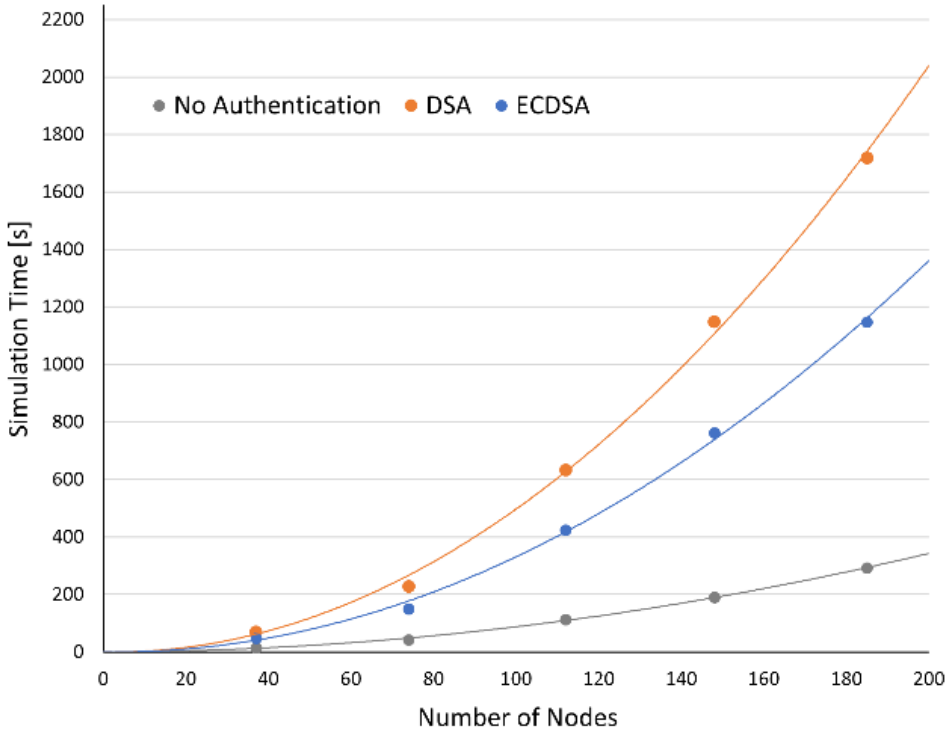

図 4.5　ノード数によるシミュレーション実行時間の変化

表 4.1　ノード数ごとのシミュレーション実行時間

Number of nodes	No Authentication [s]	DSA [s]	ECDSA [s]
37	14.84	70.53	48.08
74	43.11	227.77	149.44
112	112.51	633.83	425.37
148	190.98	1148.63	763.31
185	292.50	1719.35	1148.42

図 4.5 は，シミュレーションパターンごとのノード数によるシミュレーション実行時間の変化を近似してグラフ化したものである．具体的なシミュレーション実行時間の数値を表 4.1 に示す．これらの図表より，どのノード数の場合でも ECDSA 認証の方が DSA 認証より実効処理時間が短いことがわかる． これは，ECDSA が DSA より計算効率が良いからだと考えられる．また，ノード数が増えるほど ECDSA 認証と DSA 認証のシミュレーション時間の差は大きくなっており，ECDSA がスケーラビリティに優れていることを示唆している．

おわりに

本論文の第 1 章では, 本研究の対象である VANET や認証機構として追加したデジタル署名について概説した．第 2 章では，本研究で提案する認証機構の導入方法について述べ，第 3 章でシミュレーション環境について述べた．最後に，第 4 章でシミュレーション実験の詳細とその結果について説明した．

本研究では，V2V 通信における GPSR への認証機構追加によるルーティング性能への影響を，実在エリアのノード移動モデルにおいて調査した．その結果，不正ノードの存在するモデルでは，認証機構によって不正ノードのルーティング妨害が阻止できるとともに，配送率が向上することが確認できた．一方で，認証機構の追加は，配送率には影響しないが，通信オーバーヘッドおよび処理時間は増大させることがわかった．ただし，ECDSA は DSA に比べ，ノード数の増加に対するオーバーヘッドおよびシミュレーション時間の増加率が小さく，スケーラビリティが高いことが示された．

今後は，ノード数やノードの動き，通信のリンク状況などの要件を変更してシミュレーションを行い，多様な実環境での認証機構ルーティングへの影響を調べていきたい．また，Imghoure ら(2022)が提案している証明書なしの ECDSA を用いた場合の通信性能への影響も調べていきたい．

謝辞

本論文をまとめるにあたり，ご指導，ご示唆を賜りました岐阜大学工学部電気電子・情報工学科情報コースの三嶋美和子教授に対し深く感謝の意を示します.

また，本研究の進行中に情報交換及び議論をしていただきました岐阜大学工学部フェロー原山美知子先生はじめ，鎌部浩先生，金子美博先生，盧暁南先生，および三嶋研究室の皆様に深く感謝の意を示します.

令和 6 年 2 月 5 日

岐阜大学大学院自然科学技術研究科知能理工学専攻

階戸　弾

参考文献

[1] 間瀬憲一, ”モバイル・アドホックネットワーク(これからの情報通信とOR), ” シンポジウム(47), 13-26, 2002.

[2] D. Manivavvan, S. S. Moni, and S. Zeadally, "Secure authentication and privacy-preserving techniques in Vehicular Ad-hoc NETworks (VANETs)," Vehicular Communications 25, 100247, 2020.

[3] F. Azam, S. K. Yadav, N. Priyadarshi, S. Padmanaban, and R. C. Bansal, " A comprehensive review of authentication schemes in vehicular ad-hoc network," IEEE access, vol. 9, pp.31309-31321, 2021.

[4] L. Alouache, N. Nguyen, M. Aliout, and R. Chelouah, "Survey on IoT routing protocols: Security and network architecture," International Journal of Communication Systems 32.2, e3849, 2019.

[5] B. Ying, A. Nayak, “Efficient authentication protocol for secure vehicular communications,” 2014 IEEE 79th Vehicular Technology Conference (VTC Spring), IEEE, pp. 1-5, 2014.

[6] K. Ravi, A. S. Kulkami, "A secure message authentication scheme for VANET using ECDSA," In 2013 fourth international conference on computing, communications, and networking technologies (ICCCNT), pp. 1-6, 2013.

[7] O. S. Al-Heety, Z. Zakaria, M. Ismail, M. M. Shakir, S. Alani, and H. Alsariera, "A comprehensive survey: Benefits, services, recent works, challenges, security, and use cases for SDN-VANET," IEEE Access, vol. 8, pp. 91028–91047, 2020.

[8] A. Imghoure, A. EI-Yahyaoui, and F. Omary, "ECDSA-based certificateless conditional privacy-preserving authentication scheme in Vehicular Ad Hoc Network," Vehicular Communications 37, pp.100504, 2022.

[9] B. Karp, and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," Proceedings of the6th annual international conference on Mobile computing and networking, pp. 243-254, 2000.

[10] National Institute of Statement Technology, "FIPS 186-5, Digital Signature Standard (DSS)," February 3, 2023.

[11] National Institute of Statement Technology, "FIPS 186-4, Digital Signature Standard (DSS)," July 19, 2013.

[12] 階戸弾，原山美知子，三嶋美和子，”DSA 認証を用いた安全な V2V アドホックルーティングプロトコル，”令和五年度電気・電子・情報関係学会東海大会支部連合大会，E4-1，豊橋，2023.

[13] Dan Shinato, Michiko Harayama, Miwako Mishima, "Secure V2V Ad Hoc Routing Protocol Using Signatures," Proceedings of International Conference on Electronics, Information, and

Communication, pp.263-266, Taipei, 2024.
[14] G. F. Riley, and T. R. Henderson, "The ns-3 network simulator," Modeling and tools for network simulation, pp.15-34, 2010.
[15] 西岡正裕, 原山美知子, ”リンク品質を考慮した位置予測型 V2V ルーティングの提案, ” 信学技報, vol. 120, no. 382, pp. 35-40, 2021.
[16] A. Singh, and B. Singh, "A study of the IEEE802.11p (WAVE) and LTE-V2V technologies for vehicular communication," Automation and Knowledge Management (ICCAKM), pp. 157-160, 2020.
[17] K. G. Lim, et al.: "SUMO enhancement for vehicular ad hoc network (VANET) simulation," 2017 IEEE 2nd international conference on automatic and intelligent systems (I2CACIS), pp. 86-91, 2017.
[18] OpenStreetMap, https://www.openstreetmap.org/, 参照日:2024/1/31.

図目次

表目次