アドホックネットワークの証明書管理ノード方式における投票を用いたクラスタリング

西村　唯一郎, 長瀬　智行, 竹花　洋次郎, 吉岡　良雄

弘前大学大学院理工学研究科
〒036-8224　青森県弘前市大字文京町 3
E-mail:　nagase@eit.hirosaki-u.ac.jp

あらまし　アドホックネットワークは、インフラストラクチャを必要としない通信形態を指す。アドホックネットワークには技術的な課題が多く残されており、その中の一つに安全性の問題がある。アドホックネットワーク内で証明書を扱う公開鍵証明書分散管理方式において、ネットワーク上のノードの中から証明書管理ノードを選定し、ネットワーク全体の証明書を効率よく管理する方式として、証明書管理ノード方式が提案されている。本研究は、証明書管理ノード方式の、管理ノードの選定方法として投票を用いたクラスタリングを掲げ、その考察を行うものである。

A Voting Method based Secure Clustering for Building Certificate Management　Nodes in Ad-hoc Networks

Iichiro Nishimura, Tomoyuki Nagase, Youjiro Takehana and Yoshio Yoshioka

Graduate School of Science and Technology, Hirosaki University
3 Bunkyo-cho, Hirosaki-shi, Aomori, 036-8224 Japan
E-mail:　nagase@eit.hirosaki-u.ac.jp

Abstract: Ad-hoc networks have emerged from wireless technology, which consist of wireless nodes with absence of a stationary infrastructure. These networks are inherently vulnerable to security attacks from malicious users. Therefore, providing distinct security for ad-hoc networks becomes a primary concern and requires several challenges to be achieved. One main challenge is how to manage public keys' certificates among ad-hoc nodes. This paper introduces a secure clustering scheme to select candidate's nodes for issuing and managing public key certificates among nodes of ad-hoc network.

1. 研究背景

近年、アクセスポイントを必要とせず各ノードが無線通信でのみやりとりを行う無線アドホックネットワークの実用化に向けた研究が行われている。この方式では第三者機関が行うべき動作を各ノードが自主的に行うため、物理通信網に縛られないネットワークの構築が可能となる[1][2][3]。そのため、3 月 11 に発生した東日本大震災のような非常時での活躍などが見込まれる。

無線アドホックネットワーク通信技術は今日期待されている新しい通信技術の一つであるが、通信効率の悪さや安全性などの問題から現段階での実用化は困難である。アドホックネットワークの実現に向け、本研究では、その課題の一つである安全性に着目し、問題点と解決策について検討を行っている。

2. OLSR プロトコル

アドホックネットワーク上の冗長な通信を削減するためのアドホックネットワーク特有のプロトコルとして、OLSR (Optimized Link State Routing)がある。アドホックネットワークのルーティングプロトコルには、大きく分けてプロアクティブ型とリアクティブ型の二種類があり、中でも OLSR プロトコルはプロアクテ

ィブ型に分類される。プロアクティブ型のネットワークには、通信の度に発生する遅延を削減できるという特徴がある。

OLSR は、MPR (Multi Point Relay)集合の概念を用いて、冗長なフラッディングを減らすためのプロトコルである。通信に先だって MPR 集合を決定することによって、冗長な通信を削減することを目指している。MPR として選択されるノードは、通信速度が高ければ高いほど、電波範囲が広ければ広いほど良い。OLSR ではそのようなノードを決定するため、willingness という値を用い、再送信するノードを限定している。willingness は 0～7 の範囲の値で表される。各ノードは自身の willingness を保持し、より高い willingness を示すノードを MPR として選択することで、冗長なやりとりを削減する[4][5]。

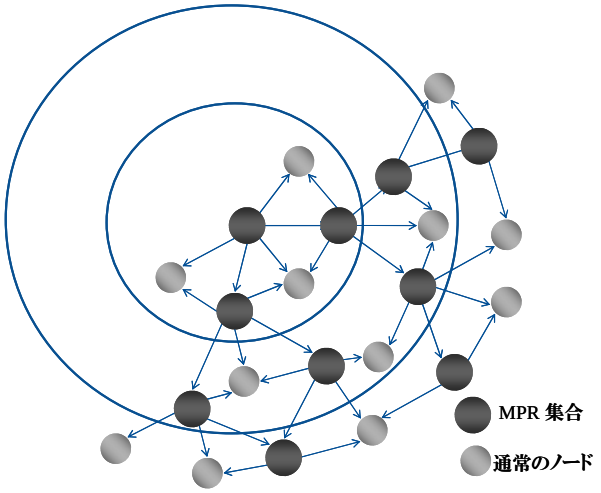

図 1　OLSR における MPR の決定

3. 公開鍵証明書分散管理方式

無線アドホックネットワーク上での通信を安全に行うための手法として、各ノードが証明書を発行し、信頼の輪を構築する、公開鍵証明書分散管理方式がある。証明書には認証者と被認証者の情報が付加してあり、各ノードはその情報を元に信頼の輪の構築を行う。

本来、公開鍵証明書を扱う場合、証明書を発行する認証局のような第三者機関が必要となる。しかし、アドホックネットワーク上にはそのような第三者機関が存在しないため、ネットワーク上の証明書は各ノードが発行・管理することになる。各ノードは自身の発行した証明書や他のノードから得た証明書を用いて、信頼の輪の構築を行う。ここで、図 2 のような偽の証明書がある場合について検討する必要がある。

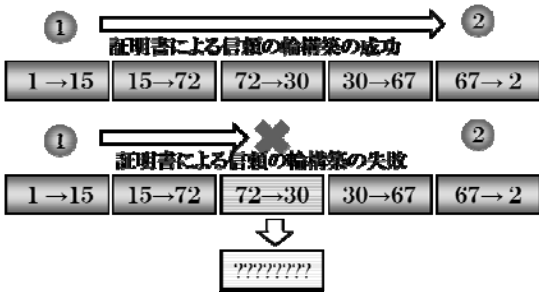

図 2　偽の証明書による認証の失敗

4. 証明書管理ノード方式

4.1. 証明書管理ノード方式の概要

通信を行うために必要な証明書を各ノードが保持し続けるのは、通信量やメモリ消費量の問題から現実的ではない。そこで、証明書管理を代行するノードを数台用意し、通信量や通信時間の削減を目指す、証明書管理ノード方式が提案された[2]。管理ノードとして選択されたノードは、自身のグループに属する一般ノードから証明書を受け取り、集めた証明書を他の管理ノードとやりとりし、同じグループの一般ノードに受け渡す。こうすることで、各ノードの保有する証明書の枚数を抑え、通信量とメモリ消費量を軽減させることができる。

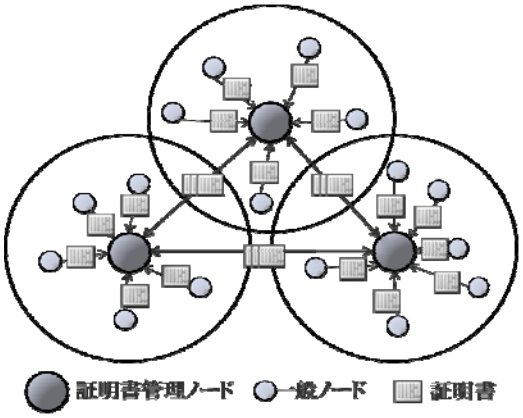

図 3　証明書管理ノード方式の概要図

4.2. 証明書管理ノード方式の特徴

ネットワーク上に 100 ノードあった場合、1 ノードにつき平均 4 ノード信頼することで、信頼の輪の構築が可能となるとされている[3]。

証明書管理ノード方式にはノードの増加に伴う大規模なリポジトリの交換が発生しないため、ノードの増加に対して有利である。また、

再クラスタリングに伴う通信量が全体の通信量の大半を占めていることから、頻繁なノードの移動に対しては不利である。これらのことから、証明書管理ノード方式は、ノードの密度が高くノードの移動速度が遅いネットワークにおいて優位性を示す。

4.3. 証明書管理ノード方式の問題点

証明書管理ノード方式では、管理ノードは willingness の値から選択する。しかし、この値は各ノードが再送信に対する積極性を示すためのものであるため、虚偽の willingness による管理ノードへの立候補を防ぐ事が出来ない。

管理ノードとなった悪意のあるノードは、証明書を改竄・破棄してしまう恐れがある。一般ノードは、自身の管理ノードとのみ証明書のやりとりを行う。そのため、自身の管理ノードが悪意を持ったノードであった場合、証明書のやりとりによる信頼の輪構築の手段を失ってしまう。また、管理ノードは他の管理ノードとも証明書のやりとりを行う。自身の管理ノードが入手した証明書が、悪意を持った管理ノードとのやりとりによって手に入れた証明書であった場合、そのグループに所属するノード全て信頼の輪構築に失敗してしまう。悪意を持った管理ノードが存在する場合、ネットワーク全体が破綻してしまう。そのため、悪意を持ったノードをネットワークから除外する、または、悪意を持ったノード以外のノードを管理ノードとするなどの工夫が必要となる。

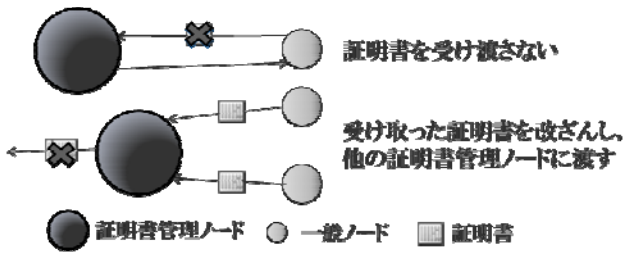

図 4　悪意を持った証明書管理ノードによる妨害の例

5. 提案方式

5.1. 投票による証明書管理ノードの決定

本研究では、信頼の輪構築の原理を応用し、管理ノードになるべきノードを投票によって決定するシステムを提案する。以下にその手順を示す。

(1)各ノードは自身が信頼するノードのノード番号を、自身の推奨ノードリストに書き込む。
(2)各ノードは自身の推奨ノードリストに書き込まれたノード番号を昇順で送信する。
(3)自身のノード番号を受け取ったノードは、その度に「自身の推奨値」を 1 増加させる。
(4)自身のノードリスト内の番号が尽きた場合、ノードリストの補充を行う。
(5)ある程度投票が済んだ段階で、各ノードは自身の推奨値のやりとりを行う。
(6)自身の推奨値が他のノードよりも高いと判断した場合、証明書管理ノードに立候補し、グループを設立する。
(7)各ノードは、立候補したノードが自身の信頼できるノードであった場合、ノードが設立したグループに参加する。

5.2. 悪意を持ったノード

悪意を持ったノードは、自身の willingness を偽ることによる管理ノードへの不正な立候補を行い、それによってネットワークを妨害することを目的としている。

各ノードは、過去に通信を行ったことがある相手や前もって公開鍵を交換し合っている相手に対して、証明書を発行することが可能である。前提として、どの正しいノードもネットワーク上のいずれかの正しいノードを最低 1 台は信頼しているが、悪意を持ったノードを信頼している正しいノードは存在しないという状況を考える。また、ネットワーク上の不正なノードを信用するノードも、同じく不正なノードであるとする。仮に複数の不正なノードが互いを信用し合っているとしても、正しいノードが信頼の輪の構築を行う際、それが不正なノードにまで及ぶことはないとする。

5.3. 推奨ノードリストと投票

推奨ノードリストは、自身が信頼するノードのノード番号を書き込むためのリストである。このノードリストは、より多くのノードから信頼されているノードを調べるため、立候補したノードが自身の信頼している相手かどうかを調べるために用いられる。ノード番号は、自身が証明書を発行した順に格納される。こうして作成されたノードリストに書かれているノード番号を、投票として昇順で発信することで、ネットワーク上のノードは、どのノードが多く

のノードに信頼されているのかを知る事が出来る。しかし、ノードリストに書き込まれたノード番号が少ない場合、その少ないノード番号を繰り返し投票することになる。そこで、ノード恒久的に投票を行うようなシステムが必要となる。

5.4. ノードリストの更新

各ノードは、自身のノードリストが尽きた場合に、自身の信頼するノードに対しノードリストの提供を依頼する。こうすることで各ノードは、自身が直接信用しないノードに対しても投票を行うことができる。

表 1 は、各ノードのノードリストと、そのノードリストから投票を行った結果であり、左上、右上、左下、右下の順で投票が行われていることを表している。表 1 の(a)に着目すると、ノード番号 0 番のノードとノード番号 10 番のノードのノードリストには、投票周期 2 の時点で投票すべきノード番号がかかれていないため、他のノードと同様に続けて投票を行うことができない。そこで、ノード番号 0 のノードは自身が信頼するノード番号 11 のノードに、ノード番号 10 のノードは自身が信頼するノード番号 1 のノードに、ノードリストの提供を依頼する。

表 1　ノードリスト更新の例(1-4 周期)

(a)

ノード番号 \ 投票周期	0	1	2	3	4	5	6
0	11	4					
1	9	0	11	8	7	3	6
2	10	3	6	9			
3	7	5	9				
4	8	7	11	1	0	9	2
5	7	11	10	0			
6	10	4	3	5	1	7	11
7	5	10	9	11	1		
8	5	0	9	3	6	2	7
9	8	3	0	2	11		
10	1	3					
11	3	5	6				

(b)

ノード番号 \ 投票周期	0	1	2	3	4	5	6
0	11	4					
1	9	0	11	8	7	3	6
2	10	3	6	9			
3	7	5	9				
4	8	7	11	1	0	9	2
5	7	11	10	0			
6	10	4	3	5	1	7	11
7	5	10	9	11	1		
8	5	0	9	3	6	2	7
9	8	3	0	2	11		
10	1	3					
11	3	5	6				

(c)

ノード番号 \ 投票周期	0	1	2	3	4	5	6
0	11	4	3	5	6		
1	9	0	11	8	7	3	6
2	10	3	6	9			
3	7	5	9				
4	8	7	11	1	0	9	2
5	7	11	10	0			
6	10	4	3	5	1	7	11
7	5	10	9	11	1		
8	5	0	9	3	6	2	7
9	8	3	0	2	11		
10	1	3	9	0	8	7	
11	3	5	6				

(d)

ノード番号 \ 投票周期	0	1	2	3	4	5	6
0	11	4	3	5	6		
1	9	0	11	8	7	3	6
2	10	3	6	9			
3	7	5	9	5	10	9	1
4	8	7	11	1	0	9	2
5	7	11	10	0			
6	10	4	3	5	1	7	11
7	5	10	9	11	1		
8	5	0	9	3	6	2	7
9	8	3	0	2	11		
10	1	3	9	0	8	7	
11	3	5	6	7	5	9	5

提供の依頼を受けたノードは、自身がどこまで提供したかを記憶しておき、最大 4 つまでノード番号を提供する。この提供の際、ノード番号に対応するノードの公開鍵も一緒に渡される。また、このノードリストは提供側の秘密鍵による認証が行われる。こうすることで、リストの提供を申請したノードは、提供されたそのリストが、自身が提供を申請した相手からの物であることを確認できる。これによって、各ノードは自身のノードリストに書かれているノード番号に対応するノード全てに、証明書を発行できる状態となる。

5.5. 管理ノードへの立候補とグループの参加

投票がある程度済んだ段階で、各ノードは自身の推奨値を他のノードとやりとりする。自身の推奨値が他のノードの推奨値よりも大きい場合、自身がより多くのノードから信頼されていると判断し、管理ノードとして立候補する。

表 2　ノードリストを用いた投票結果の例(1-4 周期)

ノード番号 \ 各ノードの推奨値	0	1	2	3	4	5	6	7	8	9	10	11
投票周期1	0	1	0	1	0	2	0	2	2	1	2	1
投票周期2	2	1	0	4	2	4	0	3	2	1	3	2
投票周期3	3	1	0	6	2	4	2	3	2	5	4	4
投票周期4	5	2	1	7	2	7	2	4	3	6	4	5

表 2 は、表 1 での投票結果をまとめたものである。この例では、最終的に 3 番と 5 番のノードが多くの票を獲得していることが分かる。仮にこの時点で推奨値のやりとりを行った場合、ノード番号 3 番と 5 番のノードは自身が管理ノードになるべきノードであると判断し、立候補を行うことになる。

立候補情報を受け取った他のノードは、そのノードのノード番号が自身の推奨ノードリスト内にあるかどうかを調べる。あった場合は、そのノードのグループに、グループへの所属を申請する。なかった場合やそのグループに所属するノードが多すぎる場合は、引き続き別の立候補情報を待つ。例えば表 1 の(d)では、ノード番号 4 番のノードは、立候補したそれらのノードを現段階で信用おらず、立候補したノードが自身の信用できるノードであることを確認するまで、そのノードのグループに所属することができない。本研究では、このように現段階でどのグループにも所属できないノードを「余剰ノード」と呼ぶこととする。

6. シミュレーション

6.1. 前提条件と動作環境

本研究では、提案方式によるクラスタリングがどのように動作するのかを観察するため、Windows XP Celeron(R) CPU 2.66GHz メモリ 502MB のパソコンを用いて C 言語プログラムを自作、実行した。本研究では、全てのノードが等しい通信速度を持ち、ネットワーク内のどのノードにもフラッディングできるという環境を想定している。

クラスタリングにおける前提条件として、4.2 節の「ネットワーク上のノード数が 100 で各ノードが平均 4 ノード信用している」という論文[3]の条件を踏襲している。また、証明書管理ノードと一般ノードの比は 1:6 が最適であるという論文[2]の結果を元に、「一つのグループに所属できる一般ノードの台数は 6 台まで」という制限と、これに関連して、「ネットワーク上の管理ノードの台数は、初期の段階で 15 台」という条件を加えている。

6.2. クラスタリングの終了条件

本シミュレーションでは、一度の投票の度に、各ノードはノードリストの補充・提供とグループ参加の判断を適宜行う。このような、投票を軸とした各ノードの一連の動作の周期を「投票周期」と呼ぶこととする。

初期段階で、各ノードは 1 ～ 7 枚証明書を発行しているため、初期段階でネットワーク上の証明書は約 400 となる。各ノードは、先述したノードリストから投票を行う動作、ノードリストを補充する動作、グループの参加を判断する動作を、一つの投票周期に一度ずつまで行う。これらの動作は、クラスタリングが終了する、すなわち余剰ノードが 0 になるまで繰り返される。本シミュレーションでは、投票周期ごとに、ネットワーク上の余剰ノードの数と証明書の枚数の変化の様子を観察している。

6.3. 仕様

本シミュレーションでは、投票周期を軸に以下のデータを記録している。

・クラスタリング終了までにかかる時間
・ネットワーク上の余剰ノード台数の推移
・ネットワーク上の証明書枚数の推移

また、これらのデータを記録するにあたり、以下のような仕様を設けている。

(1)最初に決定した管理ノードを、クラスタリング終了まで変更しない仕様。
(2)投票周期 20 周毎にグループを解体し、新たに管理ノードを選定し直す仕様。
(3)投票周期 20 周毎にグループを解体し、管理ノードの数を増やした上で管理ノードを選定し直す仕様。

6.4. 実験結果

クラスタリング終了までにかかった投票周期を、仕様(1)～(3)で比較したものを表 3 に示す。また、ネットワーク全体の余剰ノードの推移について、(1)と(2)の仕様で比較した結果を図 5、ネットワーク全体の証明書の枚数について、(2)と(3)の仕様で比較した結果を図 6 に示す。

表 3　クラスタリング終了までの仕様別の比較

グループ解体の仕様	(1)	(2)	(3)
クラスタリング終了までの投票周期平均(周目)	206.97	44.95	36.58
投票周期 255 周以内にクラスタリングが終了する確率(%)	58.8	96.2	97.6

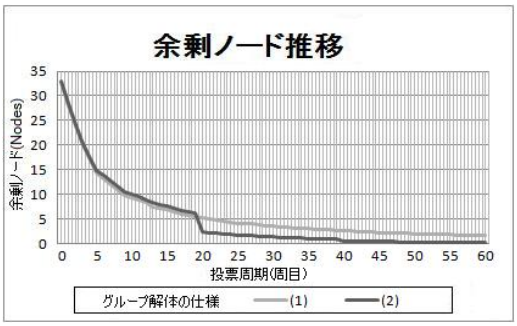

図 5　余剰ノード推移の比較のグラフ

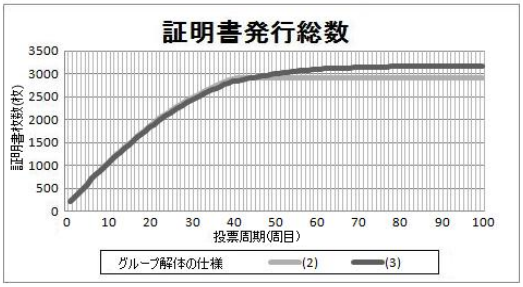

図 6　証明書総数の比較のグラフ

6.5. 考察

提案方式によるクラスタリングにおいて、どのような仕様を設けることで有用性が向上するかを考察する。動作に特別な変更を加えていない(1)のような仕様では、(2)と(3)の仕様に比べ、クラスタリング終了までの時間が極端に掛かり過ぎていることが分かる。

一度の投票周期の度に、ネットワーク上の証明書の枚数は増加し、余剰ノードの数は減少する。図 5 のグラフでは、(1)の仕様で余剰ノードが 0 にならないことが見てとれる。このことから、提案方式において、管理ノードを適宜選定し直す仕様を設けることが必須であることが分かる。また、図 6 のグラフから、管理ノードを適宜増やす仕様を採用することで、より早くクラスタリングを終了させ、ネットワーク上の証明書の枚数を抑えることができることが分かる。

7. まとめ

7.1. 提案方式のまとめ

本研究では、証明書管理ノード方式のクラスタリングにおけるセキュリティ面での弱点を指摘し、その改良案を提案方式として提示した。提案方式における動作をシミュレーションし、その結果から提案方式の性質について様々な視点から考察した。シミュレーション結果から、ネットワークの変化の様子を知ることができた。単純な仕様の変更を適用するだけでも、ネットワークを改善していることが分かった。

提案方式では、グループ参加の判定をする際にも推奨ノードリストを使用している。そのため、仮に不正な投票によって悪意のあるノードが管理ノードとなった場合でも、その管理ノードが構成するグループに正しいノードが所属することはない。提案方式ではクラスタリングの段階で不正なノードを排除しているため、クラスタリングの段階で各ノードが既存方式よりも多くの動作をすることになる。そのため、提案方式は「ノードの移動速度が遅いネットワークにおいて優位性を示す」という、既存方式の特徴を色濃く引き継いでいると言える。

7.2. 今後の課題

本研究によるシミュレーション結果は、提案方式を導入した場合の、各ノードの動作やネットワーク上の変化について考察するためのものに過ぎず、偽の証明書の排除や不正なノードの妨害への耐性を証明するためのものではない。本来であれば、各ノードの状態やネットワーク環境の変化などの、様々なファクターを考慮したシミュレーションが必要となるが、本研究ではそこまでに至っておらず、基本的な動作を検証するのみとなっている。

本研究の提案方式では、通信量やメモリ消費量を抑えることを目的とした証明書管理ノード方式に対して、各ノードの動作を増やす事でセキュリティ面の改良を目指している。本研究の有用性を証明するためには、ネットワーク環境を厳密に取り決め、既存方式よりも確実に有用であること、または提案方式が有用であるような状況があることを示す必要がある。

本研究では、基本的なクラスタリングの達成のため、いくつかの仕様を考案し、比較を行った。流動的な性質が強いアドホックネットワークでは、このような単純な仕様の変更でも、状況を見極めて行うことで非常に大きな効果を生むことが期待できる。本研究で提案した方式は改善の余地が多く残されている。今後は悪意を持ったノードの存在も加味したシミュレーションを行い、本研究の有用性を追及していく予定である。

文　献

[1] 河内 洋介, 野口 拓, 川合 誠, “送信者認証を用いた安全なアドホックネットワークルーチングプロトコル,” 電子情報通信学会論文誌, Vol.J92-B No.12, pp1844-1847, 2009.

[2] 北田 夕子, 荒川 豊, 竹森 敬祐, 渡邊 晃, 笹瀬 巌, “無線アドホックネットワークに適したルーチング情報を用いたオンデマンド公開鍵分散管理方式,” 電子情報通信学会論文誌, Vol.J88-D-1, No.10, pp1571-1583, 2005.

[3] 船曳 俊介, 磯原 隆将, 北田 夕子, 竹森 敬祐, 笹瀬 巌, “無線アドホックネットワークの公開鍵証明書管理における証明書管理ノード方式, ”情報処理学会論文誌, Vol.48, No.8, pp2835-2845, 2007.

[4] 鈴木 幹也, “アドホックネットワークにおけるOLSR制御パケットの削減法,” 早稲田大学大学院理工学研究科卒業論文, 2008.

[5] 濱口 哲志, “MPR 集合を用いたフラッディング効率化特性の評価,” 千葉大学工学部都市環境システム学科卒業論文, 2007.