センサネットワークにおける 高信頼ブロードキャストメッセージ認証技術

八百　健嗣　　福永　茂

近年，ユビキタスネットワークを代表するネットワークシステムの一つとして，無線通信機能を持つセンサを多数設置して，設備の管理や環境の観測などに役立てるセンサネットワークシステムが注目を浴びている。本稿で述べるセンサネットワークシステムは，システムを管理・制御するサーバと，低コストな多数の無線ノードから構成され，マルチホップ通信により情報をやりとりすることを想定する。

我々は，このセンサネットワークシステムにおいて，無線ノードの障害を修復したり，機能を変更・追加したりするために，各ノードに搭載するソフトウェアを無線経由で更新する技術を開発している。この技術により，設置された多数のノードを回収する手間なく，サーバが各ノードから情報を収集して，各ノードの状況に応じたソフトウェアを，各ノードに持たせることが可能となる。たとえば，施設に設置されているセンサネットワークシステムを有効活用するために，業務営業期間中は温度情報を収集するシステムとして，業務休業期間中は不正侵入を検知するシステムとして動作させたり，無線ノードとデジタル情報家電機器とを，必要に応じて直接通信させたりできる。

ここで，考慮すべき課題の1つに，セキュリティがある。無線通信を利用することから，攻撃者が容易に不正な更新データをネットワークに投入できる。ここで，もしノードが不正な更新データを排除できないと，センサネットワークシステムは攻撃者の思うままに乗っ取られてしまう可能性がある。この脅威を図1に示す。一方，ソフトウェアを更新する場合，各ノードが正常に更新を完了したことを確認することが重要となる。特に，更新データの配布の過程においては，配布した更新データが確実に全ノードに行き届いたことを確認する必要がある。以上のことから，更新データの配布は，次の2つの要件を満たす必要があると考えられる。

- 各ノードが受け取ったデータを，各ノードがサーバからの正しいデータであると認証すること
- サーバが発信したデータが，正しく全ノードに届いたことをサーバが確認すること

本稿では，センサネットワークのネットワーク構造の一例であるマルチホップツリー構造において，上記2つの要件を満たすメッセージ認証方式を提案する。

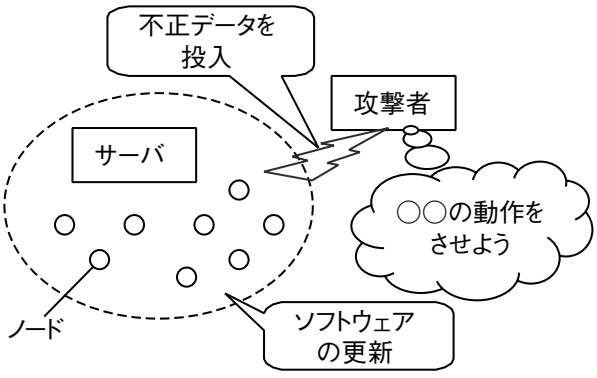

図1　無線通信を用いたソフトウェア更新の脅威

ノードのハードウェア制約と求められるメッセージ認証方式

センサネットワークを形成する無線ノードは，低コストを重視して開発される。したがって，メッセージ認証は，低コストなCPUでも動作するように，計算量が少ないアルゴリズムを用いて実現することが望ましい。また，ノードは小型で可搬性があるため，攻撃者がノードを盗み，内部メモリに格納されている鍵などの情報を不正に読み出す可能性がある。特に，低コストを重視して開発されるノードには，コスト高となる耐タンパ性メモリ（鍵などの秘密情報の漏洩を物理的に困難にするメモリ）を搭載するとは限らないため，内部メモリに格納されている鍵情報は簡単に読み出されてしまうことを考慮する必要がある。したがって，メッセージ認証は，ノードが保持する鍵情報が漏洩しても堅固な方式であることが望ましい。

一般的に，メッセージの認証には，認証子を利用する。メッセージの送り手がメッセージに対する認証子を生成・

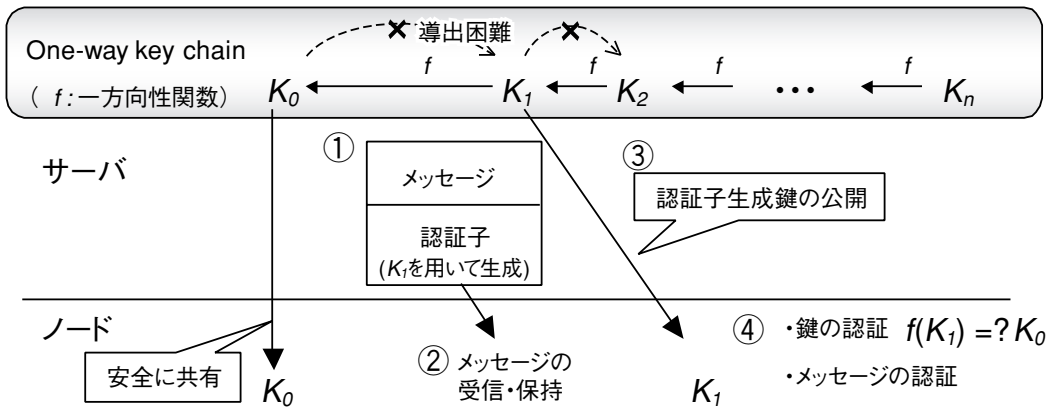

図2　One-way key chainを用いたメッセージ認証の動作例

付加して送信し，メッセージの受け手がその認証子を検証することで，受け取ったメッセージがメッセージの送り手からの正しいメッセージであることを確認する。認証子の生成・検証方式は，共通鍵暗号を用いる方式と，公開鍵暗号を用いる方式に大別される。

共通鍵暗号を用いたブロードキャストメッセージ認証は，サーバと全ノードに予め共通の鍵を持たせ，その鍵でメッセージに対する認証子を生成・検証することで実現する。共通鍵暗号を用いた方式は，公開鍵暗号を用いた方式と比較して，計算量が少ないという特徴がある。しかし，サーバと全ノードが共通の鍵を持っているため，ノードはサーバを厳密には認証できない。たとえば，攻撃者がノードを盗んで内部メモリに格納されている共通鍵を不正に入手すると，攻撃者はサーバになりすました不正なメッセージをノードに認証させることが可能になる。

それに対して，公開鍵暗号を用いたブロードキャストメッセージ認証は，サーバが秘密鍵でメッセージに対する認証子を生成し，ノードは公開鍵で認証子を検証することで実現する。公開鍵暗号を用いた方式では，秘密鍵と公開鍵が異なり，特定の秘密鍵で生成した認証子の検証は，その秘密鍵に対応する公開鍵でしか成功しない。また，公開鍵から秘密鍵を求めることは計算量的に困難であることから，公開鍵（ノードが認証子の検証に用いる鍵）を秘密にする必要がないという特徴がある。しかし，公開鍵を用いた認証子の検証処理は，共通鍵暗号を用いた認証子の検証処理と比較して計算量が二桁程度大きく，特に処理能力が低いノードにとっては，大きな負荷となってしまう。公開鍵暗号の計算コストを抑えて，処理能力が低いノードへの搭載を目指す研究も実施されているが，実用的な計算量には達していないのが現状である[1)]。

センサネットワークシステムにおけるサーバのメッセージ認証には，共通鍵暗号方式のように計算量が少ないアルゴリズムを用いながらも，公開鍵暗号方式のようにノードが秘密情報を持たずにサーバからのメッセージを認証できるような方式が望まれる。

One-way key chainを用いたメッセージ認証

従来方式として，共通鍵暗号を用いたメッセージ認証でありながら，ノードが秘密情報を持たずにメッセージを認証できる，One-way key chainを用いたメッセージ認証方式がある[2)]。

One-way key chainとは，ランダムな値に一方向性関数を複数回施して生成された鍵鎖列である。一方向性関数とは，入力値から出力値を求めることは容易だが，出力値から入力値を求めることは困難な性質をもつ関数をいう。One-way key chainを用いたメッセージ認証では， One-way key chainの各鍵をメッセージに対する認証子の生成鍵として用いる。

図2を用いて，One-way key chainを用いたメッセージ認証の手順を示す。

＜初期設定＞

サーバと全ノードは，システムで規定する一方向性関数 *f* を把握している。サーバはOne-way key chainを生成し，秘密に保持する。ノードは，サーバが生成したOne-way key chainの最後の値 *K0* を鍵情報として安全に保持する。

① サーバは，One-way key chainの鍵を，生成と逆順で使用する。サーバはOne-way key chainの次の鍵 *K1*（まだ各ノードに公開していない鍵）でメッセージに対する認証子を生成し，送信する。

② ノードは，認証子付きのメッセージを受信する。この時点では，受信したメッセージを認証することはできず，メッセージを保持しておく。

③ サーバは，①でメッセージの認証子生成に用いたOne-way key chainの鍵K_1を送信（公開）する。

④ ノードは，受信した（公開された）鍵K_1に一方向性関数fをかけて，予め保持しているOne-way key chainの鍵K_0と一致するかどうかを確かめる。もし一致するならば，受信した鍵K_1をサーバが生成したOne-way key chainの公開された鍵であるとし，保持する。次に，鍵K_1を用いて②で受信したメッセージの認証子を検証する。検証が成功することで，メッセージをサーバからの正しいメッセージであると認証する。

以上のようにして，ノードはサーバからのメッセージを認証する。サーバが次にメッセージを送信する時には，まだノードに公開していないOne-way key chainの次の鍵K_2でメッセージに対する認証子を生成し，送信する。

この方式においてノードが保持する鍵情報は，サーバが次に公開するOne-way key chainの鍵を認証するための情報であり，秘密ではない。もし，攻撃者がノードの保持する鍵情報を不正に入手したとしても，その鍵情報から次にサーバがメッセージの認証子生成に用いる鍵を求めることは，一方向性関数の原理により困難である。

マルチホップ通信環境における脅威

上述した方式は，サーバが公開する鍵情報を先に知るノードが，後に知るノードに対してサーバへなりすますことができるという欠点を持つ。たとえば，マルチホップ通信環境におけるルータノードは，伝達遅延を故意に発生させることによりサーバへなりすますことができる。不正なルータノードによるサーバへのなりすましの例を**図3**に示す。不正なルータノードは，サーバが発信したメッセージを次ホップのノードへ中継せず，サーバがそのメッセージの認証子生成鍵を公開するまで待つとする（**図3**(i)）。そして，鍵が公開された時に不正メッセージにその鍵で認証子を付与し，次ホップのノードへ伝達する（**図3**(ii)）。後に，既に公開されている認証子生成鍵をまるで今サーバによって公開されたかのように次ホップのノードへ伝達することで，中継先の全ノードに対して不正なメッセージを認証させることが可能になる（**図3**(iii)）。

図3で説明したようなサーバへのなりすまし攻撃を防ぐためには，サーバが鍵を公開した後に生成されたメッセージを，各ノードが無効であると判断できる仕組みが必要となる。この仕組みを実現する方式の一つに，サーバと全ノードが時刻同期をとる方式がある[2)]。この方式では，One-way key chainの各鍵が認証子生成鍵として有効である時間をサーバと全ノードとが把握しておくことによ

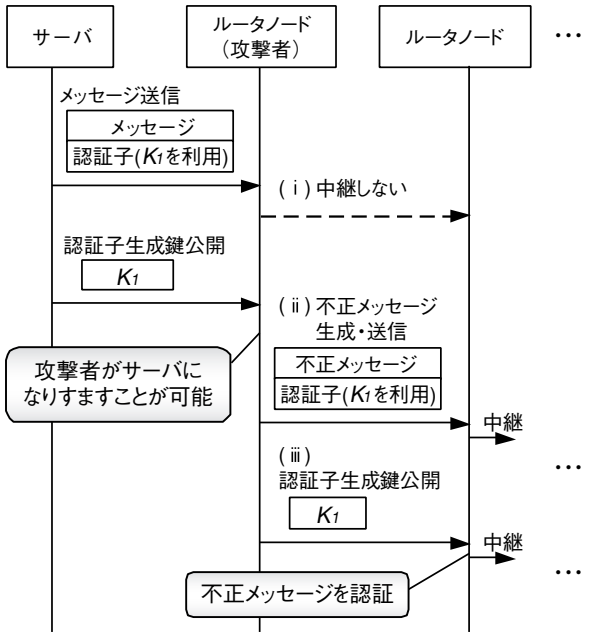

図3　不正なルータノードによるサーバへのなりすまし

り，サーバが鍵を公開した後に生成されたメッセージを，各ノードが無効であると判断する。この方式において，攻撃者が不正メッセージをノードに認証させることができるかどうかは，サーバと全ノードが偽りなく時刻同期を行っているかどうかに依存する。この時刻同期をどのようにして実現・維持するかは，別途大きな課題となる。

高信頼ブロードキャストメッセージ認証

マルチホップ通信環境においても，攻撃者によるサーバへのなりすまし攻撃に耐性があり，かつメッセージの受信確認を得ることを特徴とする，高信頼ブロードキャストメッセージ認証を提案する[3)]。提案方式の要点は以下の2つである。

- サーバは，認証させたいメッセージが全ノードに偽りなく届いたことを確認した後で，メッセージの認証子生成に用いた鍵を公開する。
- One-way key chainの各鍵で認証するメッセージ数を規定し，サーバとノード間で認証の回数同期を行う。

提案方式において，攻撃者が不正なメッセージをノードに認証させることができるかどうかは，メッセージの受信確認を偽れるかどうかに依存する。そこで，セキュアな受信確認を実現する一例として，マルチホップツリー構造においてメッセージの受信確認を効率良くかつセキュアに収集する方式を提案した[4)]。この方式では，各ノードはサーバと1対1の固有鍵を共有することを前提とする。そして，メッセージを受信したノードは，その固

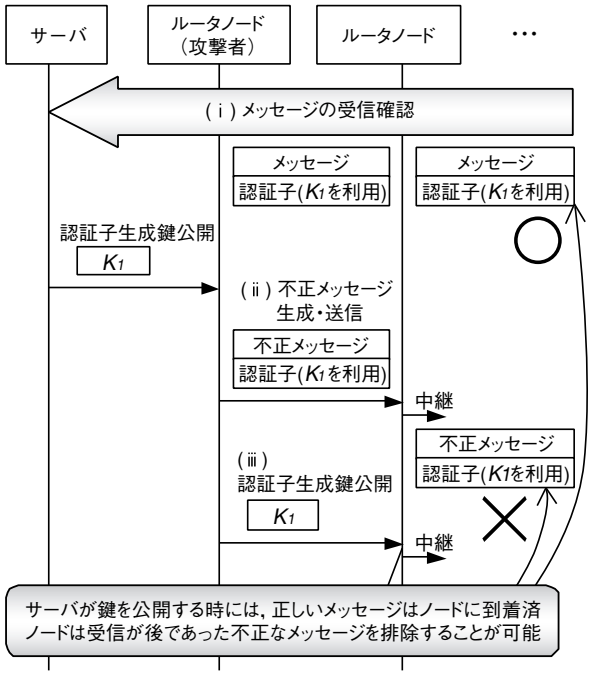

図4　メッセージ受信確認後の鍵公開

有鍵を用いてメッセージに対する受信確認情報を生成する。受信確認情報の返信では，ルータノードごとに自分と全ての子ノードの受信確認情報を圧縮しながらサーバへ返信することによって，受信確認を返信することで生じる通信オーバヘッドを抑制し，かつ受信確認情報の偽造を困難にできる。

このように，サーバが全ノードに対してメッセージの受信を確認した後で，鍵を公開することによって，各ノードがその後に生成された不正なメッセージを無効であると判断できるようになる。**図4**を用いて動作を説明する。まず，サーバが全ノードからメッセージの受信確認を得る（**図4（i）**）。この時点で，全ノードには正しいメッセージと認証子の組が到達していることになる。不正なルータノードは，サーバへなりすますために，サーバが公開した認証子生成鍵で不正メッセージに認証子を付与し，次ホップのノードへ伝達する（**図4（ii）**）。後に，既に公開されている認証子生成鍵を次ホップのノードへ伝達する（**図4（iii）**）。しかし，この時点では，全ノードに正しいメッセージと認証子の組が到達済みである。もし，サーバの公開した認証子生成鍵で，1つのメッセージだけを認証することが規定されているとすると，ノードはメッセージを受信した順序により，受信が後であった不正なメッセージを排除することが可能となる。

同じく，One-way key chainの各鍵で複数のメッセージを認証するように規定する場合は，それら全てのメッセージの受信確認を得た後で，認証子生成鍵を公開する。 もし，攻撃者が公開された鍵を用いて不正メッセージをネットワークに投入したとしても，ノードは受信が後であった不正メッセージを，規定数より多いメッセージとして排除できる。

ま と め

以上，マルチホップ通信環境においても，サーバへのなりすまし攻撃に耐性があり，かつ受信確認を伴った高信頼ブロードキャストメッセージ認証方式を提案した。提案方式は，一般的に計算量が少ない共通鍵暗号系を用いて実現し，また鍵を不正に読み出された時の堅固性を考慮しているため，低コストを重視して開発される無線ノードから形成されるセンサネットワークシステムにおいても動作可能であると考える。この技術を成熟させることにより，状況に応じてネットワークの動作を変更したり，機能を追加したりする，柔軟性を持ったセンサネットワークシステムをセキュアに構築することが可能となる。

今後は，より柔軟性の高いメッセージの受信確認方式の検討と，実機への実装による評価が課題となる。 ◆◆

■参考文献

1）G.Gaubatz,*et al.*：“Public key cryptography in sensor networks-revisited,” 1st European Workshop on Security in Ad-Hoc and Sensor Networks （ESAS2004）.
2）A. Perrig, *et al.*：“SPINS: Security Protocols for Sensor Networks,” Wireless Networks J., vol.8, no.5, pp.521–534 (2002).
3）八百，松村，福永：“センサネットワークにおける高信頼ブロードキャストメッセージ認証方式”， 情報処理学会研究報告，2005-CSEC-28，pp.241–246，2005年
4）八百，川本，松村，福永：“センサネットワークのマルチホップツリー構造に適したセキュアな受信確認方式”，情報処理学会研究報告，2004-MBL-30， pp.69–75，2004年

●筆者紹介

八百健嗣：Taketsugu Yao. 研究開発本部 ユビキタスシステムラボラトリ
福永茂：Shigeru Fukunaga. 研究開発本部 ユビキタスシステムラボラトリ