Vol.2013-MBL-65 No.37
Vol.2013-UBI-37 No.37
2013/3/15

センサネットワークにおける グループ鍵管理の通信量について

楫 勇一[1]　野田 潤[2]

概要：大規模センサネットワークでは，複数のセンサノードによってネットワーク内部にグループを構成し，グループ単位で情報の流通を制御する機会も多い．グループ鍵を利用することで，グループを単位とした情報の流通制御やデータ認証等を効率的に実現可能であるが，安全な運用のためには，グループ鍵を効率的に更新する仕組みの実現が不可欠である．著者らは，センサネットワーク特有のグループ構造に着目し，グループ鍵更新のための枠組みとして属性ベースグループ鍵管理方式（ABGKM）を提案している．実験的評価により，ABGKM の通信量は LKH 等既存のグループ鍵更新プロトコルに比べて小さくなることが示されているが，ABGKM における制御可能なパラメータが全体の通信量に与える影響については，十分解明されているとは言えなかった．本研究では，ABGKM の鍵更新における諸事象を確率的モデルで記述し，グループ鍵更新に必要となる通信量を決定する計算式を導出する．この計算式から定まる値は実験的評価の結果とよく整合しており，ABGKM の性能予測や動作の最適化等に応用することができる．

1. はじめに

本論文では，センサネットワークにおけるグループ鍵管理プロトコルの通信量について議論を行う．ある程度の規模を持つネットワークにおいては，複数のユーザや端末等により**グループ**を構成し，グループを単位として，情報の流通やシステムの管理・運用を行うことも多い．とくに，機密性の高い情報を取り扱うネットワークにおいては，グループを構成するユーザや端末である**メンバ**と，メンバ以外のユーザ・端末とを峻別することが重要となる．情報や資源が一箇所に集中して存在する場合，ある種のアクセス制御機構を設けて不正なアクセスを遮断することも可能であるが，情報そのものがネットワーク上を流通する場合は，情報を暗号化し，グループのメンバのみに**グループ鍵**と呼ばれる暗号鍵（復号鍵）を配布しておくことが有効である．グループ鍵の利用により，大規模なグループベースのサービスや管理を安全かつ効率的に実現することが可能となるが，システムを安全に運用するためには，グループ鍵を適切に管理し，更新することが必要となる．すなわち，新規メンバの加入や既存メンバの脱退等によってグループのメンバ集合が変化したときには，それまで使われていたグループ鍵を無効化し，新しいものに置き換えなければならない．ここで注意しなければならないのは，グループを脱退するメンバは，自分が脱退した後に使われる新しいグループ鍵を入手することを目的とし，不正な行為を行う可能性があるという点である．グループ鍵の更新を安全に実現する手法については，暗号や情報セキュリティの分野で長く研究が行われており [2], [3], [6]，たとえば LKH 法 [7], [8] のように，インターネット上での大規模グループでのグループ鍵管理を想定し，安全性と効率性とを両立する方式等が多数提案されている．

センサネットワークにおいてもグループの概念は有効であり，グループ鍵についても，その利用価値は高い．センサネットワークにおけるグループ鍵の管理において，LKH 等，既存のグループ鍵管理方式を採用することも考えられるが，センサネットの環境に特化し，より簡便で軽量な仕組みを模索する研究も行われている [1], [5], [9], [10]．一方，筆者らは，これとは少し異なる視点から検討を行なっている．詳細は後述するが，センサネットワークにおいて想定される「グループ」と，LKH 等において想定されている「グループ」との間には，ある種の質的な差異があると考えられる．すなわち LKH 等で想定されているグループは自律的に行動するユーザにより構成される制御不能な集団であるのに対し，センサネットワークにおけるグループは，システム設計者により意図的に構成される制御可能なノード集合であると考えることができる．この特性を利用

[1] 奈良先端科学技術大学院大学
Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 630-0101, Japan
[2] 日本電気株式会社
NEC Corporation, 1753 Shimonumabe, Nakahara-Ku, Kawasaki 211-8666, Japan

© 2013 Information Processing Society of Japan

することにより，LKH 等の一般的なグループ鍵管理方式よりも，効率の良い方式を開発することができると考えられる．この考え方に基づき，筆者らは，**属性ベースグループ鍵管理方式**（Attribute-Based Group Key Management, 本稿では ABGKM と略す）を提案している [4]．ABGKM では，ノードの集合分割の族により複数のグループを定義する．グループ鍵の更新においては，安全性の損なわれていないグループ鍵を利用したグループ通信を有効に活用し，効率的にグループ鍵の更新を行うことができる．

LKH, ABGKM を含め，これまでのグループ鍵管理方式の研究では，送信されたデータが確実にノードの手元に届くことを前提条件として，ある種の最適化がなされている．実際の通信システムでは，たとえば通信の過程でデータ消失が発生し，データの再送処理が必要になる場合も多いが，そのような「例外的処理」に必要となる通信量については，従来のグループ鍵管理方式の研究においては考慮されていない．すなわち，従来研究の多くは，プロトコルスタックの高位レイヤのみに着目して議論を行なっており，そのレベルでの最適化が実際の通信量削減にどの程度寄与するのか，明確でない点があった．

この問題に取り組むため，筆者らは，ABGKM におけるグループ鍵の配送データに，「多重度」と呼ばれるある種の冗長性を付与する方式を検討している．この方式では，鍵を管理するサーバは，グループ鍵の更新に必要となるデータをいくつかの異なった鍵で暗号化し，その結果得られる複数の暗号データをネットワークに送出する．グループのメンバであるノードは，送出された暗号データの 1 個でも受信・復号できればグループ鍵を正しく更新することができるため，たとえ他の暗号データが手元に届かなかったとしても，再送処理を行う必要はない．この方式では，最初に送出すべきデータ量は増加するものの，再送処理等に必要となる通信量を削減することが可能となっている．したがってデータの消失が頻発する環境や，再送処理に大きな通信コストが発生するような環境では，全体として通信量を削減することが可能になると期待される．実際，ネットワークシミュレータにより計算機実験を行ったところ，多重度を適切に制御することにより，全体の通信量を削減できることが確認されている．

本研究は，それら実験的な結果に対し，数理的な立場から考察を与えるものである．ABGKM における鍵更新プロトコルの振る舞いを確率的モデルで記述し，グループ鍵更新に必要となる通信量を導出するための計算式を求める．この計算式から定まる値は実験的評価の結果とよく整合しており，ABGKM の性能予測や動作の最適化等に応用することができる．

2. 属性ベースグループ鍵管理方式

2.1 属性ベースグループ

一般には，グループは，センサノードにより構成される任意の集合として定義される．しかし，実際のセンサネットワークにおいてグループを無作為に構成することは現実的ではなく，通常は，ネットワーク管理者の判断に基づき，何らかの共通の性質を持つノードをまとめて一つのグループとすることが一般的である．また，1 つのネットワークの中に複数のグループを構成したり，1 台のノードが複数の独立したグループに所属するようなことも，十分に一般的な形態として考えることができる．そのようなグループの汎用的なモデルとして，本研究では，**属性ベースグループ**の概念を導入する．**属性**とは，ノードに対して定義された何らかの性質や特性であり，たとえば「設置場所」「メーカ種別」「搭載されるセンサの種別」「MAC アドレスの最上位バイト」等，様々な属性を自然発生的に，あるいは人工的に定義することができる．一般性を失うことなく，すべてのノードは，それぞれの属性に対して唯一の**属性値**を持つと考えることができる．もちろん，属性を不用意に定義してしまうと，ある属性について 1 台のノードが複数の属性値を持つような状況も発生しうるが，その場合は，「属性値の集合の部分集合族」を値とするような属性を別途考えることで，属性値の一意性を確保することができる．また，特定のノードに対して属性値が定義できないような場合は，「未定義」という特別な属性値を導入すればよい．以上のような属性値の一意性を仮定したうえで，本研究では，ある属性に対して同一の属性値を持つノードの集合を，1 つのグループと考えることにする．

以下の議論では，ノードすべてからなる集合を N と表記し，属性を，N の集合分割であると定義する．

定義 2.1 N の集合分割 $A = \{G_1, \ldots, G_m\}$ を**属性**と呼ぶ．すなわち，$G_j \subset N$, $j_1 \neq j_2$ のとき $G_{j_1} \cap G_{j_2} = \emptyset$ であり，$G_1 \cup \cdots \cup G_m = N$ である． □

直感的には，G_j は，この属性について j 番目の属性値を持つノードの集合に相当する．いま，属性が全部で d 種類存在し，$i(1 \leq i \leq d)$ 番目の属性が $A_i = \{G_{i,1}, \ldots, G_{i,m_i}\}$ として与えられるものとする．このとき，各 $G_{i,j}(1 \leq i \leq d, 1 \leq j \leq m_i)$ を**基本集合**と呼ぶ．

定義 2.2 以下のように定義されるノードの集合 G を，r 次の**属性ベースグループ**と呼ぶ．

$$G = G_{i_1,j_1} \cap \cdots \cap G_{i_r,j_r}. \tag{1}$$

ただし $1 \leq i_1 < \cdots i_r \leq d$ とし，$1 \leq c \leq r$ である c に対して $1 \leq j_c \leq m_{i_c}$ とする． □

上記の定義において $r = 1$ とすることにより，基本集合は，1 次の属性ベースグループであるということもできる．

属性ベースグループは，属性の種類に対して指数的な個

© 2013 Information Processing Society of Japan

数だけ定義されることになる．その中には，実用上意味のないノードの集合も含まれる可能性があるが，属性の選択が適切であれば，センサネットワークにおいて実用的に意味のあるノードの集合は，属性ベースグループのいずれかになっていると考えることができる．たとえば，「2013 年に 2 階に設置されたノードの集合」を定義したい場合は，対象となるノードの集合でもって任意のグループを構成するのではなく，「設置年」と「設置場所」という属性を導入し，その属性ベースグループとして，目的となるノードの集合を規定してやることができる．これらの属性の導入に伴って，たとえば「1900 年に地下 120 階に設置されたノードの集合」のような意味のない属性ベースグループが定義される可能性もあるが，そのような属性ベースグループは無視してよい．それらのグループはあくまでも形式的に定義されるだけのものであり，後述するグループ鍵管理の効率や安全性に影響を与えることはない．以上の考察に基づき，以下の議論では，属性ベースグループを単に**グループ**と呼ぶことにする．

何を属性とし，どのような属性値を導入するかは，センサネットワークの用途や必要となる要求仕様に応じて決定する必要がある．属性構造の設計には大きな自由度があるが，本研究では，以下に定義する「完全性」を満足するよう，属性と属性値が定められることを仮定する．

定義 2.3 属性の集合 $A_1,\ldots,A_d$ が**完全**であるとは，任意の d 次グループが 2 個以上のノードを含まないときをいう． □

この条件は，すべての属性について属性値が完全に一致するようなノードは，ネットワーク内に 2 台以上存在しないということを意味している．通常，センサネットワークに設置されるノードにはそれぞれに固有の役割があるため，すべてのノードは互いに異なった部分があると考えられる．それらノードの役割に着目して属性を適切に選べば，上記の完全性を満たすことは容易であると考えられる．

例 2.1 図 1 に，完全な属性集合の例を示す．ただし $N=\{n_1,\ldots,n_7\}$ とし，

$$G_{1,1}=\{n_1,n_2,n_3\}, G_{1,2}=\{n_4,n_5\}, G_{1,3}=\{n_6,n_7\},$$
$$G_{2,1}=\{n_1,n_4\}, G_{2,2}=\{n_2,n_5,n_6\}, G_{2,3}=\{n_3,n_7\},$$
$$G_{3,1}=\{n_1\}, G_{3,2}=\{n_2,n_4\}, G_{3,3}=\{n_3,n_5,n_6,n_7\}.$$

である．たとえば，3 次のグループである $G_{1,1}\cap G_{2,1}\cap G_{3,1}$ を考えると

$$G_{1,1}\cap G_{2,1}\cap G_{3,1}=\{n_1\}$$

であり，その要素数は 1 であることが確認できる．その他の 3 次グループについても，要素数は必ず 0 か 1 となることから，この属性集合が完全であることがわかる． □

次節ではグループ鍵更新の手順について述べるが，そこでは，基本集合によるノードの**集合被覆**の概念が必要となる．ただし，本研究で考える集合被覆は，計算量理論等で考えられている古典的な集合被覆を一般化したものとなっており，以下のように定義される．

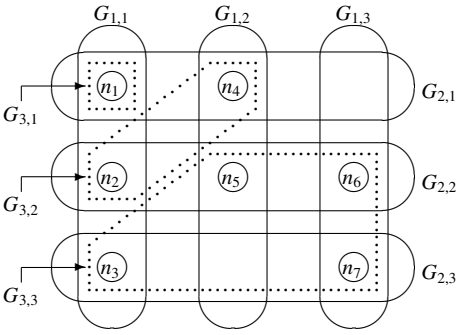

図 1 完全な属性集合の例

定義 2.4 m を正整数，$n\in N$ をノードとする．基本集合の族 $\mathcal{G}=\{G_1,\ldots,G_l\}$ が以下の条件を満たすとき，$\mathcal{G}$ を (m,n)-**集合被覆**と呼ぶ．

(1) 任意の $n'\in N\setminus\{n\}$ に対し，$\mathcal{G}$ は，n' を要素として含む基本集合を m 個以上含む．
(2) すべての $i(1\le i\le l)$ について $n\notin G_i$ である． □

$\mathcal{G}$ が (m,n)-集合被覆であれば，n 以外の任意のノードは，「少なくとも m 回，$\mathcal{G}$ の要素である基本集合により（古典的な意味で）被覆される」ことになる．以下では，上記の m を**多重度**と呼ぶ．先ほどの例 2.1 においては，$\mathcal{G}=\{G_{1,2},G_{2,2},G_{2,3},G_{3,2},G_{3,3}\}$ が $(2,n_1)$-集合被覆となっている．たとえばノード n_3 は $G_{2,3}$ と $G_{3,3}$ により被覆されており，定義 2.4 の最初の条件を満たすことがわかる．n_3 以外の任意のノードについても同様に，少なくとも 2 回被覆されていることを確認できる．ただし，ノード n_1 だけは $\mathcal{G}$ のどの基本集合にも被覆されておらず，したがって，被覆から「排除」されていると考えることができる．

2.2 グループ鍵と鍵の運用

前節で定義した属性ベースグループに対し，グループ鍵を定義することを考える．本方式式におけるグループ鍵は，鍵の管理者が任意に選択して与えるものではなく，ある種の基本的な情報から導き出されるものとなっている．

基本集合 $G_{i,j}(1\le i\le d, 1\le j\le m_i)$ に対し，**基本鍵** と呼ばれる秘密情報 $k_{i,j}$ を割り当てることを考える．基本鍵 $k_{i,j}$ は，$G_{i,j}$ に属するノードのみに対して安全に配布されているものとする．属性ベースグループに対するグループ鍵は，以下の手順により基本鍵から計算される．

定義 2.5 属性ベースグループ G が式 (1) のように与えられるとする．このとき，G の**グループ鍵**を $k(G)=h(k_{i_1,j_1}\|\cdots\|k_{i_r,j_r})$ と定義する．ここで h はハッシュ関数であり，"‖" は基本鍵の連接を表す． □

この定義から明らかなとおり，あるノード n がグループ鍵 $k(G)$ を計算できるのは，$n\in G$ のとき，かつ，そのときの

みである．

グループ鍵の管理・運用のため，基本鍵には**世代番号**と**使用開始時刻**, **使用終了時刻**が対応付けられているものとし，これらをまとめて鍵の付随情報と呼ぶ．以下では，基本鍵には必ず付随情報が付与されているものとし，基本鍵の送受や記録の操作について述べる際に，付随情報について言及しない場合もある．鍵の管理者（以下では，この管理者を**鍵管理サーバ**と呼ぶ）は，新しいノードをネットワークに設置する際，その時点で有効な基本鍵（および付随情報）をノード内部の安全な領域に格納する．すなわち，ノード内部における鍵の初期化操作は，オフラインで安全に実行されているものとする．

各ノードには，ある程度の耐タンパ性を有する記憶装置が備えられているものとし，ここに基本鍵（および付随情報）を格納するものとする．ただし，装置の耐タンパ性について万全の安全性を仮定することは現実的でないため，本研究では，「ノードに対する攻撃がなされてから，耐タンパ装置に格納された情報が漏洩するまでの時間」が「ノードが攻撃された事実を鍵管理サーバが知りうるまでの時間」よりも長いことを前提条件として仮定し，この条件の下で安全な方式を検討する．各ノードは，上記の耐タンパ装置の中に，全部で $2d$ 個の基本鍵を格納する．$2d$ 個の基本鍵のうち d 個は**現行基本鍵**と呼ばれ，残り d 個は**次期基本鍵**と呼ばれる．現行基本鍵は現在使用されている基本鍵であり，次期基本鍵は，現行基本鍵が使えなくなったときに使用する予定の基本鍵である．各ノードは，以下のルールに従って，これら基本鍵の管理および運用を行う．

- 鍵管理サーバから送信される鍵更新情報を受信し，「次期基本鍵」を計算して耐タンパ装置に格納する．
- 「次期基本鍵」の使用開始時刻になると，「現行基本鍵」を「次期基本鍵」で置き換える．
- 「現行基本鍵」の使用終了時刻が近づいているにも関わらず「次期基本鍵」が登録されていない場合は，鍵管理サーバに対し，鍵更新情報の（再）送信を要求する．

2.3 鍵更新プロトコル

グループ鍵は，主として 2 つの理由により更新が必要となる．第 1 の理由は，一つの鍵を長期間にわたって使い続けるのを避けるためである．鍵漏洩のリスクを抑え，攻撃者に暗号解読の手がかりを与えないためには，定期的にグループ鍵の更新を行う必要がある．グループ鍵の更新が必要となる第 2 の理由は，グループメンバの構成変更に対応するためである．センサネットワークの長期運用においては，故障したノードやバッテリが切れたノードを新しいノードと交換したり，あるいは，なんらかの理由で紛失したノードを無効化したり，まったく新規にノードを追加設置するような作業も発生する．この場合，それまでグループのメンバであったノードをグループから除外し，新しいノードをグループに追加する作業が必要となる．グループから除外されたノードを物理的に回収できる場合，ノード内部に格納された基本鍵を安全に消去することも可能であるが，ノードの回収が困難な場合，とくに，ノードが盗難等にあった場合は，ノード内部の機密情報が悪意を持った部外者に漏洩する危険性が生じる．すなわち，第 2 の理由によるグループ鍵の更新においては，グループから除外しようとしているノードの内部に格納されている基本鍵が安全でないことを前提とし，鍵更新の手順を考える必要がある．

以下では，ノード $n \in N$ が盗難に遭った状況を想定し，基本鍵の更新を行う手順について議論する．一般性を失うことなく，ノード n は d 個の基本集合 $G_{1,1}, \ldots, G_{d,1}$ に属するものと仮定する．この場合，不正者は d 個の基本鍵 $k_{1,1}, \ldots, k_{d,1}$ を入手している可能性があり，$k_{i,1} (1 \le i \le d)$ を継続して使用すると，機密情報が不正者に知られるおそれがある．ネットワークを安全に運用するためには，鍵管理サーバが新しい基本鍵 $k'_{i,1}$ を決定し，$G_{i,1} \setminus \{n\}$ に属する全てのノードに $k'_{i,1}$ を配布する必要がある．$G_{i,1} \setminus \{n\}$ が比較的小さければ，新しい鍵をユニキャスト的に配布することも可能である．しかし，一般には $G_{i,1} \setminus \{n\}$ は多数のノードを含む可能性があり，また，$1 \le i \le d$ なるすべての $G_{i,1} \setminus \{n\}$ に対して情報を配布することを考えると，より効率的でスケーラビリティの高い方式を検討することが望ましい．

ここでは，次期基本鍵を無作為に選択するのではなく，一方向的な手順を用いることで，現行基本鍵から次期基本鍵を生成することを考える．ただし，鍵の生成を無条件に行えたのでは，盗難にあったノード n も新しい鍵を入手できてしまう．これを防ぐため，一方向的な計算に，副次的な情報である**ソルト**を利用することを考える．ソルト鍵管理サーバが選ぶランダムな情報であり，n 以外のすべてのノードに対して配布される．これにより，n 以外のすべてのノードにおいてすべての基本鍵を更新することができ，その一方，ノード n のみが新しい基本鍵を知り得ない状況を作り出すことができる．上記のような状況を効率的に作り出すため，以下では，ソルト暗号化し，マルチキャスト，またはブロードキャストすることを考える．

鍵管理サーバが行うべき手順は，以下のとおりである．

(1) 多重度 m とソルト s を決定する．また，新しい基本鍵の世代番号 v'，使用開始時刻 a' および使用終了時刻 e' を決定する．

(2) (m,n)-集合被覆 $\mathcal{G} = \{G_{i_1,j_1}, \ldots, G_{i_l,j_l}\}$ を計算する．ここで l は，$\mathcal{G}$ を構成する基本集合の個数である．

(3) $1 \le c \le l$ のそれぞれに対し，以下の情報をマルチキャスト，あるいはブロードキャストする．

$$M_c = (i_c, j_c, E(k_{i_c,j_c}, s \| v' \| a' \| e')). \quad (2)$$

ここで $E(k,x)$ は，鍵 k で情報 x を暗号化して得られる暗号データを表す．

© 2013 Information Processing Society of Japan

M_c を受信したノード n' は，以下の手順を実行する．

(1) M_c の第 1 要素，第 2 要素から i_c, j_c を特定する．$n' \notin G_{i_c,j_c}$ の場合は M_c を破棄し，以下の手順は実行しない．$n' \in G_{i_c,j_c}$ の場合は次のステップに進む．

(2) M_c の第 3 要素を復号し，s, v', a' および e' を入手する．

(3) 以下のいずれかを実行する．

- もし次期基本鍵が登録されていない場合は，このノード n' が所属する各基本集合 $G_{i,j}$ について，$h(k_{i,j} \oplus s)$ を $G_{i,j}$ の次期基本鍵として登録する．
- 次期基本鍵が既に登録されており，その世代番号が v' よりも小さい場合は，現在の次期基本鍵を破棄し，上記の手順と同様にして次期基本鍵を登録する．
- 上記のいずれにも相当しない場合は，鍵情報の更新を行わない．

ノード n' は，最大で m 個の M_c を通じてソルト s' を入手する可能性がある．このため，鍵の世代番号を参照し，鍵生成の一方向操作を重複して行わないようなっている点に注意されたい．

この方式の特徴は，ネットワークに送出される情報 (2) が，(m,n)-集合被覆に基づいて決定される点である．これにより，n 以外の任意のノードに対し，少なくとも m 個，そのノードが復号可能な暗号データを送信することになる．通信途中のデータ消失等により，これら m 個のデータのいくつかは消失してノードにまで到達できない可能性もあるが，m 個のうち 1 個のデータでもノードの手元に届けば，そのノードでは鍵の更新を行うことが可能となる．すなわち，最初に冗長性を付与した形でデータを送信することにより，鍵更新に失敗した場合に発生する再送操作を回避していると考えることができる．

ここで述べたグループ鍵の定義，鍵更新手段を総称して，**属性ベースグループ鍵管理方式**（Attribute-Based Group Key Management, ABGKM）と呼ぶ．

3. 計算機実験

前節で述べた ABGKM では，鍵管理サーバから最初に送出する情報に冗長性を与えることによって再送に必要となる通信コストを削減し，全体の通信量を抑えることを狙いとしている．冗長性の大きさは多重度 m により制御されるが，m の値を必要以上に大きくしてしまうと，再送コストの削減量よりも，冗長性付与に伴うオーバヘッドが大きくなる恐れもある．ABGKM の有効性について検討し，m の適切な値を求めるため，ネットワークシミュレータを利用して通信量の予備的評価を行った．

ここでは簡単のため，2.3 節で述べた最初の理由，すなわち，鍵の定期更新を行う際の通信量について評価する．第 2 の理由による鍵更新，すなわち，あるノードをグループから除外するシナリオを考えるためには，かなり多くの前提条件等を導入する必要があり，評価結果について明確にならない恐れがあるためである．

本実験では，1,024 台のノードが 32 台 ×32 台の正方グリッド上に配置された無線センサネットワークを考える．グリッドの一辺は 8 メートルとし，ネットワーク全体では 248 メートル四方にノードが配置されることになる．グリッドには x 座標と y 座標が割り振られており，座標 (x,y) に設置されたノードには，$x+32y$ として決まるノード番号が与えられるものとする．座標 $(15,15)$ に配置された 495 番のノードは，ネットワーク全体のほぼ中央に位置することになるため，このノードを鍵管理サーバとして設定する．

このネットワークにおいてグループを構成するため，10 個の属性 $A_1,\ldots,A_{10}$ を定義する．本来であれば，これら属性はネットワークの設置目的や各ノードの性質等に基づいて決定すべきところであるが，ここでは人工的な例を考えているため，ある種，機械的に属性と属性値とを定義することにする．具体的には，$A_i = \{G_{i,0}, G_{i,1}\} (1 \le i \le 10)$ であり，基本集合 $G_{i,j}$ は「ノード番号を 2 進表記したとき，上から i ビット目が j になるノードの集合」とする．たとえば，ノード番号が $858 = (1101011010)_2$ であるノードは，

$$G_{1,1}, G_{2,1}, G_{3,0}, G_{4,1}, G_{5,0}, G_{6,1}, G_{7,1}, G_{8,0}, G_{9,1}, G_{10,0}$$

の 10 個の基本集合に所属することになる．

ノードはグリッド上に配置されているため，この例では，一つの行，一つの列を構成する 32 台のノードが，それぞれ 1 個のグループを構成していると考えるのが自然である．すなわち，第 1 行グループから第 32 行グループ，第 1 列グループから第 32 列グループの，全部で 64 個のグループがネットワークに存在すると考える．このようにして定義されるグループは，属性ベースグループの一つである．たとえば，$y = 0$ の第 1 行グループは，ノード番号 $0 = (0000000000)_2$ から $31 = (0000011111)_2$ までの 32 台のノードにより構成される．これは，5 次の属性ベースグループ $G_{1,0} \cap \cdots \cap G_{5,0}$ と一致しており，上述のように機械的に定義した属性ベースグループの中に，実用上意味のあるグループが定義されていることがわかる．

上記のようなノード配置とグループ構成とを想定し，グループの鍵を更新するのに必要となる通信量を評価した．ネットワークシミュレータとしては Qualnet を利用する．ネットワーク層プロトコルとしては IPv4, MAC 層および物理層のプロトコルとしては IEEE 802.15.4 を用いるものとする．物理層ペイロードは 127 バイトであり，このサイズは，式 (2) の情報 M_c を 1 個のパケットに格納するのに十分な大きさである．したがって，M_c 1 個を送信するのに 1 パケットが必要であると考える．無線通信には 2.4GHz 帯を用い，O-QPSK 変調にて 250kbps の通信速度を仮定する．送信電力やアンテナ利得等については，電波の到達距離が 14 メートル程度となるよう調整している（具体的には，自由空間伝搬モデルを仮定し，送信電力を −17dbm, ア

© 2013 Information Processing Society of Japan

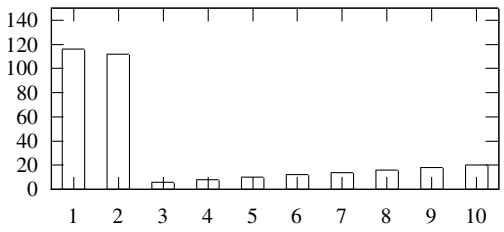

図 2 鍵管理サーバが送信するパケット数

ンテナ利得を −3.0dB に設定した）．電波の到達距離が 14 メートルあるため，1 台のノードは，自分を中心として 8 方位にある 8 台の隣接ノードと直接通信することが可能である．ノード間の通信経路については，ツリー状の静的な経路を手動で設定した．

鍵管理サーバは 200ms の間隔でパケットを送信するものとし，各ノードは，ランダムジッタを利用してパケットの衝突を回避することを仮定した．ネットワーク層におけるブロードキャスト制御については，経路上での親ノードが子ノードの動作を監視し，passive ACK により再送制御を行う．ただし，passive ACK の最大待ち時間は 150ms に設定し，ネットワーク層での再送は最大 3 回までとした．

プロトコル実行開始時点においては，すべてのノードに現行基本鍵を格納しておく．ただし現行基本鍵の使用終了時刻を 60 秒後とし，次期基本鍵は登録しない．この状態から鍵更新のプロトコルを開始し，鍵管理サーバがソルトを送信するものとする．復号可能な情報を 1 個でも受信したノードは次期基本鍵の取得に成功するが，60 秒を経過しても有効な情報を受信できないノードは，鍵管理サーバに対して再送要求を送信する．再送要求のメッセージも消失する可能性があるため，ノードは，1 秒おきに再送要求の送信を繰り返すものとする．ただし，再送回数に上限は設けず，次期基本鍵が正しく取得できるまで，この動作は繰り返される．再送要求が鍵管理サーバに到達した場合，サーバは，ユニキャスト通信によりソルトを再送する．

シミュレーションにおいては，多重度の値を 1 から 10 まで変化させながら，鍵更新が完了するまでのパケット数を記録した．多重度と，サーバが送信したパケット総数との関係を示すシミュレーション結果を図 2 に示す．横軸が多重度 m であり，縦軸は，サーバが送信したパケットの総数である．ただし，このパケット総数の中には，ノードからの再送要求への応答に必要なパケットの個数も含む．グループ鍵の定期更新においては，鍵管理サーバは最初に，$G_{i,j}(1 \le i \le m, j = 0, 1)$ それぞれに対応して合計 $2m$ 個のパケットを送信する．もし全てのノードが 1 個以上の復号可能な暗号データを受信できれば，鍵の再配送は全く必要ないため，通信パケット数の総数は $2m$ だけでよい．図 2 のグラフでは，$m \ge 3$ の各多重度に対してパケット数が $2m$ と一致しており，3 以上の多重度では，全てのノードが再送なしで鍵更新に成功していることがわかる．一方，多重度が 1 または 2 の場合は，復号可能な暗号データを受信できないノードが発生してしまう．それらのノードは再送処理を要求し，鍵管理サーバは，ソルトを再送するために新たな通信トラヒックを発生させることになる．グラフにおいて $m = 1, 2$ のときにパケット数が増加するのは，この再送処理のための通信量が大きいためである．

表 1 受信失敗ノード数と再送要求数

多重度 m	受信失敗ノード数	再送要求数
1	13	396
2	6	289
3–10	0	0

表 1 に，多重度のそれぞれの値について，復号可能な暗号データを受信できなかったノードの台数，および，それらノードが送信した再送要求メッセージの個数を示す．図 2 のグラフからもわかるとおり，$m \ge 3$ のときは受信に失敗するノードが存在しないが，m が 1 または 2 のときは，受信に失敗するノードが存在していることがわかる．これらのノードは再送要求メッセージを鍵管理サーバに送信するが，そのタイミングは，現行基本鍵の使用終了時刻付近に集中するためネットワークに輻輳現象が生じることになり，再送要求メッセージが鍵管理サーバまで届かない事象も少なからず発生する．すなわち，1 回の再送を実現するための通信コストが，かなり大きくなっていることが理解できる．

4. 通信量の数式モデル

前節では，多重度を変化させることにより，実際の通信量が変化することを計算機シミュレーションにより確認した．通信量を削減するためには最適な多重度の値を決定する必要があるが，これが実験的にしか行えないのであれば，実用上は好ましくない．通信環境を取り巻く条件はネットワークにより異なるうえ，時々刻々と変化する場合も考えられるためである．そこで本節では，多重度に対する通信量の変化の様子を数式モデルで表すことを考える．この数式モデルは，ネットワークにおけるパケットの紛失確率をパラメータとして含むものとなっており，たとえば，鍵管理サーバがパケット紛失確率を観測・推測することにより，通信環境の変動に追随して多重度を動的に変化させるような対応も可能になると考えられる．すなわち，数式モデルの確立は，実用上の運用改善にも寄与することが期待できるのである．

ここでは，鍵管理サーバを根とするツリートポロジによりネットワークが構成されるものとし，ツリーの高さを h_m と表記する．このツリートポロジにおいて，深さ $h(0 \le h \le h_m)$ の位置にあるノード，すなわち，鍵管理サーバとの通信に h 段のマルチホップが必要となるノードの集

© 2013 Information Processing Society of Japan

合を N_h と書くことにする．ノード間の通信は互いに独立であると仮定し，1 ホップの通信において，確率 p でパケットの消失が発生すると仮定する．この場合，サーバが送信した 1 個のパケットが，サーバから h ホップ離れたノードに到達して受信される確率は $(1-p)^h$ であり，パケットが途中で消失する確率は $1-(1-p)^h$ である．

ABGKM では，1 台のノードに対して m 個以上，復号可能なパケットが送出される．このうち 1 個でもノードに到達すれば，そのノードは鍵の更新に成功するが，m 個すべてが消失すると，鍵更新ができないことになる．パケット消失の事象が互いに独立であると仮定すると，m 個のパケットすべてが消失する確率は $(1-(1-p)^h)^m$ により与えられる．サーバから h ホップ離れた位置にいるノードの集合が N_h により与えられるため，鍵管理サーバからの最初のパケット送出で鍵更新に失敗する（復号可能な暗号データを全く受信できない）ノード台数の期待値は，

$$\sum_{h=1}^{h_m} |N_h|(1-(1-p)^h)^m \tag{3}$$

となる．これが，全ノードの中で再送処理を必要とするノードの数となる．

再送処理において，鍵管理サーバは，ノードに対してユニキャスト的に情報の送信を試みる．1 回のユニキャスト通信（1 個のパケット送信）で，N_h に属する 1 台のノードがソルトの受信に成功する確率は，さきほどの議論と同じく $(1-p)^h$ により与えられる．これに失敗した場合，サーバは 2 回目のユニキャスト通信を試みるが，その 2 回目のユニキャストで情報の送受が完了する（すなわち，最初に送信されたパケットが消失し，二回目に送信されたパケットが無事に到着する）確率は $(1-p)^h(1-(1-p)^h)$ となる．一般に，k 回目のパケット送信でデータ送受が完了する確率は $(1-p)^h(1-(1-p)^h)^{k-1}$ であり，したがって，h ホップだけ離れたノードに対して，データを確実に配送するために必要となるパケット送信回数の期待値は

$$r_h = \sum_{k=1}^{\infty} k(1-p)^h(1-(1-p)^h)^{k-1} = 1/(1-p)^h \tag{4}$$

により与えられる．

式 (3) および (4) より，再送処理においてサーバが送信するパケット総数の期待値は

$$\sum_{h=1}^{h_m} r_h|N_h|(1-(1-p)^h)^m$$

と見積もることができる．一方，最初のブロードキャストの際に送信されるパケットの個数，すなわち，多重度 m の集合被覆のサイズが $g(m)$ により与えられるとすると，サーバが鍵更新操作全体を通じて送信するパケットの総数は

$$f(m) = g(m) + \sum_{h=1}^{h_m} r_h|N_h|(1-(1-p)^h)^m$$

$$= g(m) + \sum_{h=1}^{h_m} |N_h|(1-(1-p)^h)^m/(1-p)^h$$

となる．この関数 $f(m)$ の最小値を与える m が，サーバの送信パケット数を最小化する多重度となる．したがって，$f(m)$ を微分し，その導関数が 0 となる m を求めることにより，最適な多重度を決定することが可能となる．

上記の関数 $f(m)$ は一般的な形で与えられており，その具体的な値や極小点を決定するためには，関数 $g(m)$ およびノード集合 N_h の値を具体的に定める必要がある．以下では，上記数式の妥当性について検証するため，前節で行った計算機実験に類する環境を想定して $g(m)$ および N_h を具体的に定め，上式についての評価を行う．前節の計算機実験では，全部で 10 個の属性を想定し，各属性には 2 個の属性値を定義した．このシナリオにおいてグループ鍵の定期更新を行う場合，多重度 m の集合被覆のサイズは $g(m) = 2m$ となる．一方，議論を簡単にするため，ネットワークのトポロジやルーティングについては多少の近似を行い，次数 8 の完全木によるツリートポロジ（したがって $|N_h| = 8^h$ となる）を想定し，最大ホップ数が $h_m = 4$ であると仮定する．この場合，$|N_1| = 8, |N_2| = 64, |N_3| = 512, |N_4| = 4096$ となり，ノード台数は全部で 4680 台となる．以上の条件のもと，上記の数式 $f(m)$ は

$$f(m) = 2m + \sum_{h=1}^{4} 8^h(1-(1-p)^h)^m/(1-p)^h$$

として具体的に与えられる．この $f(m)$ の値を図 3 に示す．ここで横軸は多重度 m であり，データ消失確率 p は 0.001, 0.01, 0.025, 0.05, 0.075, 0.1 の 6 通りとしている．この図からわかるとおり，$f(m)$ は下に凸の形状をしており，ある特定の m に対し $f(m)$ は最小値を取る．最小値を与える m の値はパケット消失確率 p に依存しており，p が大きくなるにしたがって，$f(m)$ の極小点（最小点）は図中の右上方向に移動する．このことは，通信路の品質が悪い環境下では，多重度を大きく取って送信データの冗長性を上げることが全体の通信効率の最適化に有効であることを示している．たとえば，$p = 0.001$ の場合，パケット消失の頻度が非常に小さく，各ノードは，サーバの送信したパケットを非常に高い確率で受信できると期待される．この場合，多重度をあげてブロードキャストパケットの個数を増やすことは得策でなく，単にオーバヘッドを増加するという悪影響を生じるのみである．一方，$p = 0.05$ 程度になるとパケット消失の確率も無視できず，多重度が小さすぎると，ブロードキャストパケットによる鍵更新に失敗するノードが少なからず発生してしまう．この場合は，多重度 m を 4 または 5 程度に設定して鍵更新を行うことで，再送対応の発生を未然に防ぐことが適当であると判断できる．多重度を増やすことはブロードキャスト時の通信量を増加する方向に働くが，多数発生するであろう再送処理に多くの通信量を費

© 2013 Information Processing Society of Japan

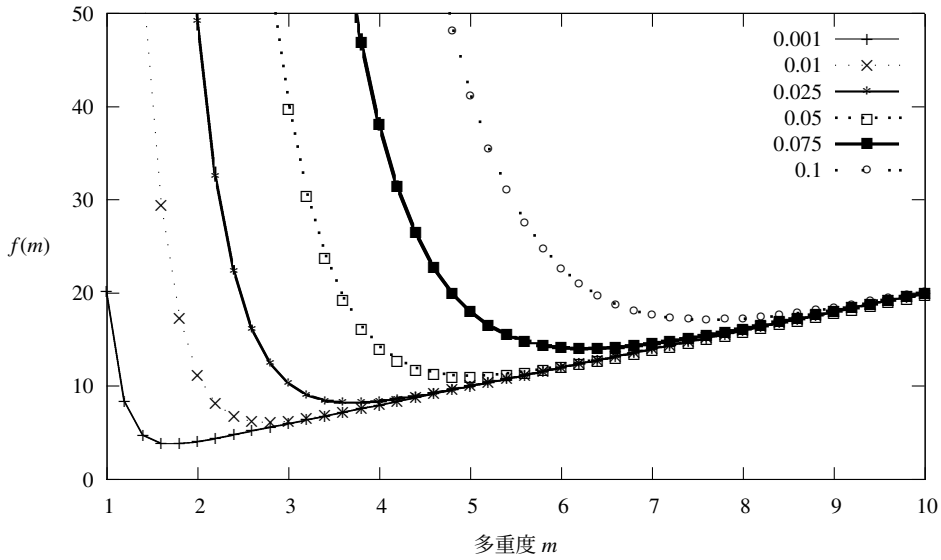

図 3 多重度と通信量の関係

やすよりは，最初のブロードキャスト時に冗長性を持ったデータ通信を行うほうが，全体的には効率を改善することができる．

この数式モデルを利用して多重度を決定するためには，ネットワークにおけるパケット消失確率 p を見積もる必要がある．p の値を正確に特定することは困難であると考えられるが，たとえばノードからの ACK メッセージの個数等に関する統計量を解析することにより，パケット消失確率について，ある程度の見積もりを与えることは可能であると予測される．鍵管理サーバにおいて，運用上の工夫と本節で導出した数式とを用いることにより，グループ鍵更新に必要となる通信量を削減することが可能となる．

5. まとめ

本研究では，ABGKM の振る舞いを確率的モデルで記述し，グループ鍵更新に必要となる通信量を導出するための計算式を求めた．この計算式の示す結果は計算機実験の結果とよく一致しており，ABGKM の最適な運用に貢献することができると考えられる．

謝辞 本研究の一部は，総務省「最先端のグリーンクラウド基盤構築に向けた研究開発（省電力アクセスネットワーク制御技術）」の研究支援により行われた．

参考文献

[1] R. Dutta, E. Chang, and S. Mukhopadhyay, Efficient Self-Healing Key Distribution with Revocation for Wireless Sensor Networks Using One Way Key Chains, 2007 Applied Cryptography and Network Security, pp. 385–400, 2007.

[2] B. Jiang and X. Hu, A Survey of Group Key Management, 2008 Intl. Conf. on Computer Science and SoftwareEng., Wuhan, China, pp. 994–1002, 2008.

[3] E. Jung, A. Liu, and M. Gouda, Key Bundle and Parcels: Secure Communication in Many Groups, Computer Networks, 50, pp. 1782–1798, 2006.

[4] J. Noda, Y. Kaji, and T. Nakao, A Group Key Management Scheme for Sensor Nodes Belonging to Multiple Large-Scale Groups, Journal of Information Processing, *52*, 3, pp. 1160-1172, 2011 (in Japanese).

[5] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler, SPINS: Security Protocols for Sensor Networks, Wireless Networks, *8*, 5, pp. 521–534, 2002.

[6] S. Rafaeli and D. Hutchison, A Survey of Key Management for Secure Group Communication, ACM Computing Surveys, *35*, 3, pp. 309–329, 2003.

[7] S. Setia, S. Zhu and S. Jajodia, A Comparative Performance Analysis of Reliable Group Rekey Transport Protocols for Secure Multicast, Special Issue of Performance Evaluation, pp. 21–41 Performance 2002, Rome, Italy, 2002.

[8] C.K. Wong, M. Gouda, and S.S. Lam, Secure Group Communications Using Key Graphs, 1998 ACM SIGCOMM Conf. on Applications, Technologies, Architectures, and Protocols for Comput. Comm., pp. 68–79, 1998.

[9] S. Zhu, S. Setia, and S. Jajodia, LEAP: Efficient Security Mechanism for Large-Scale Distributed Sensor Networks, 10th ACM Conf. on Comput. and Comm. Security, pp. 62–72, 2003.

[10] IEEE 802.15 WPAN Task Group 4 (TG4), http://www.ieee802.org/15/pub/TG4.html ⟨10.04.2012⟩

© 2013 Information Processing Society of Japan