INSA de Lyon

rapport PIR

La gestion des clés dans les réseaux de capteurs

Auteurs :
Corentin LE DEVEDEC

Superviseur :
Walid BECHKIT

Article :
Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks
Auteurs :
Mohamed F. Younis, Senior Member, IEEE
Kajaldeep Ghumman
Mohamed Eltoweissy, Senior Member, IEEE
Date :
Août, 2006

Mars, 2017

1 Introduction

Ces dernières années, il y a eu des avancées majeures dans le développement des micros capteurs sans fil, de faible puissance. Tout d'abord qu'est ce qu'un réseau de capteurs à grande échelle ? C'est un système qui est composé d'innombrables capteurs autonomes qui sont souvent regroupés en groupes ou clusters. Ces capteurs ne sont toutefois pas complètement autonomes puisqu'ils nécessitent des passerelles et de puits, qui centralisent toutes les informations recueillies par les différents capteurs, afin de pouvoir utiliser les architectures informatiques déjà en place. Chaque groupe nécessite donc une passerelle dédiée qui permettra de transférer les données collectées, à l'aide des capteurs, par l'intermédiaire de canaux de communication sans fil jusqu'au puits. L'émergence d'une telle technologie a encouragé la création de réseaux de capteurs à grandes échelles à plusieurs usages, dans les domaines militaires et civils. Les applications sont donc multiples comme la surveillance des champs de batailles, la gestion de catastrophes naturelles ou encore le taux de pollution dans les villes. Ces applications, qui ne sont que quelques exemples parmi tant d'autres, soulèvent certains problèmes, comme ceux de l'autonomie énergétique, des protocoles de communication ou encore de la sécurité. En effet la sécurité émerge comme un défi majeur à cause de l'absence de protection physique de ces systèmeset du fait de transmissions des communications sans fil.

2 Contexte et Problématique

Ces capteurs sans fil disposent de faibles capacités de mémoire, d'énergie et de calcul, aussi rendre ce type de réseaux complètement autonomes et sécurisés nécessite d'implémenter des protocoles d'optimisation de la consommation de l'énergie et de sécurité pour éviter que des «intrus», en s'introduisant dans le réseau, falsifient, suppriment, récupèrent illégalement des données sensibles, ou rendent le réseau inopérable. Cependant les limites en ressources, énergie et mémoire, imposées par les capteurs rendent la tâche d'incorporer un protocole de sécurité difficile.

3 Schéma de base

Le protocole initial est hiérarchique, c'est à dire qu'il aboutit à une gestion évolutive des clefs et des communications sécurisées multigranularité. Cette solution est aussi distribuée, c'est à dire qu'elle répartit les différentes activités de la gestion des clés entre les nœuds multiples du réseau. Elle emploie la méthodologie du Système de Base d'Exclusion, une solution qui permet de générer et de rafraîchir efficacement les clefs dans les grands réseaux. Une solution proposée pour sécuriser les réseaux à micros-capteurs sans fil, le protocole SHELL, est présentée dans [1] L'organisation d'un réseau utilisant le SHELL est basée sur une architecture composée de plusieurs clusters, qui regroupe une multitude de capteurs physiques et d'une passerelle. Les capteurs sont chargés de collecter des données qui seront transmises via la passerelle à la station de controle.

4 Prédéploiement

Le SHELL est basé sur un système basique d'exclusion (EBS, Exclusion Basic System) proposé dans [2]. EBS est une méthodologie d'optimisation combinatoire pour la gestion des clefs. En se basant sur un ensemble de k+m cléfs initial le système EBS permet de construire un ensemble de N combinaisons uniques (sous-ensembles uniques) de k cléfs. A cause de l'unicité des sous-ensembles, le nombre maximal de sous-ensembles est de $C = \binom{k+m}{k}$. Un tel protocole nécessite que $\binom{k+m}{k}$ soit supérieur ou égal au nombre N de nœud. Pour construire un EBS de paramètres (N, k, m) avec des valeurs de k et m possibles, il faut déterminer le nombre de manières possibles de former un sous-réseau de k nœuds parmi un groupe de (k+m) nœuds. Par exemple, la matrice canonique A de l'EBS (10 ; 3 ; 2) contient l'énumération de toutes les façons $\binom{5}{3} = 10$ de former un sous ensemble de trois clés à partir de cinq clés, comme indiqué dans le tableau 1 ci-dessous :

On peut donc conclure que le pré-déploiement est déterministe et que les clefs qui sont déjà chargées sont les clefs de communication entre les passerelles et le puits, ainsi que les clefs de découvertes des capteurs lors de l'initialisation du réseau après déploiement.

	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9
K_1	1	1	1	1	1	1	0	0	0	0
K_2	1	1	1	0	0	0	1	1	1	0
K_3	1	0	0	1	1	0	1	1	0	1
K_4	0	1	0	1	0	1	1	0	1	1
K_5	0	0	1	0	1	1	0	1	1	1

FIGURE 1

5 Détails du protocole SHELL

Tout d'abord il est important de comprendre que chaque nœud possède une base de données contenant les identifiants des autres nœuds et que chaque nœud doit donc stocker K clefs administratives, avec K le nombre initial de clefs car K = k + m, et m clefs de communications/-cryptages. Chaque passerelle est génératrice d'une matrice EBS. En effet un EBS est généralement constitué d'un cluster. Ainsi il suffit d'une passerelle par EBS pour créer la matrice canonique de l'EBS qui génère les clefs pour son cluster. Les passerelles peuvent aussi rafraîchir sur demande les clefs, détecter et évacuer les nœuds compromis. La mise en route de ce protocole commence immédiatement après le déploiement des capteurs dispersés dans une zone géographique. Cette mise en route du protocole se décompose en trois grandes parties : la mise en place du réseau, la génération des clefs de sécurité et enfin la création des clusters :

- Le protocole commence par un enregistrement des passerelles, pour ce faire celles-ci établissent un canal de communication avec la station de controle en envoyant un message broadcast par lequel la passerelle indique de manière cryptée son identifiant et sa géolocalisation à l'aide d'une clef qui leur a été attribuée au préalable. Une fois les connexions puits-passerelle établies, le puits envoie des clefs spécifiques afin que les connexions inter-passerelles soient créées, ainsi que d'autres clefs nommées Ksg, pour la phase de reconnaissance des capteurs.
- Une fois les connexions inter-passerelles établies, la phase de détection des capteurs peut commencer. Les capteurs diffusent en cas de besoin, leur identifiants et leur géolocalisation aux passerelles qu'ils auront au préalable chiffré avec une clef de communication. Lors de la réception de l'annonce envoyée par un capteur, une passerelle déchiffrera le message grâce à une clef qu'elle partagera avec ce capteur et rentrera son identifiant et son emplacement dans sa base de données. Dès que la détection du capteur est terminée, les capteurs génèrent une nouvelle clef Ksg à l'aide d'une fonction de hashage unidirectionnel. Pour des raisons de sécurité, cette fonction est connue e la station de controle ainsi que des capteurs mais pas des passerelles. La finalité de cette opération, est de rendre obsolète les clefs Ksg actuelles connues des passerelles. C'est un processus crucial pour la récupération d'une passerelle compromise.
- Puis il y a la création des clusters, c'est-à-dire que les capteurs doivent être regroupés en clusters, les passerelles collaborent alors entre elles pour former ces clusters. Chaque cluster est géré par une passerelle. Pour la méthode d'EBS, la portée et l'emplacement géographique sont des critères utilisés pour regrouper les nœuds en clusters en accord avec les références données dans les publications [3]et [4]

Ensuite les différentes matrices EBS sont crées. Pour ce faire, chaque passerelle effectue une analyse des clusters qui lui sont rattachés pour définir le nombre de clés administratives nécessaires au bon fonctionnement des clusters. EBS propose des compromis en ce qui concerne les exigences de stockage. Une augmentation du nombre K de clefs administratives, augmente la capacité de stockage par nœud, tandis qu'une augmentation de nombre de clefs de cryptage, C, conduit à un surcoût de la communication des nouvelles clés en cas de régénération des cléfs. En fonction de la taille du cluster et de la mémoire disponible dans le capteur, la passerelle optimise la valeur des paramètres k et m d'un EBS. Pour un grand cluster, il peut être préférable de diminuer la valeur de k pour limiter les besoins de stockage par nœud. Ceux sont deux des critères de sélection des valeurs des variables k et m pour la formation des matrices. Une fois ces matrices formées, certaines passerelles seront affectés à la tête des différents EBS pour être les passerelles génératrices des clefs de sécurités car on le rappelle, les clefs qui sont stockées au préalable dans les passerelles et les nœuds sont des clefs de découvertes qui ont pour principal objectif de permettre d'établir les connexions élémentaires dont a besoin le SHELL pour initialiser sa mise en route. Une fois que la station de controle aura annoncé ses caractéristiques, ces passerelles vont générer les m clefs, nécessaires au cryptage des données. Ces clefs

sont utilisées par les noeuds pour communiquer entre eux. Le fonctionnement normal de ce protocole est assez simple, car une fois la mise en place effectuée, il se contente de rafraîchir périodiquement les clefs et d'ajouter d'éventuels nouveaux capteurs.

6 Amélioration du protocole

Les auteurs du protocole SHELL [2]ont souhaité améliorer la prévention des attaques avec complicité (des "collusions" en anglais). Deux nœuds colludent lorsqu'ils partagent les clefs les uns avec les autres. Dans le cas du protocole SHELL, les mêmes clefs étant utilisées dans plusieurs nœuds, les probabilités d'échanges d'information par des nœuds compromis augmentent, ce qui peut aboutir à la compromission d'un grand nombre de clefs et éventuellement lacapture du réseau. C'est un problème récurrent dans les allocations de clefs. Le schéma directeur pour diminuer les risques est d'exploiter la proximité des nœuds afin qu'un nœud ne partage un maximum de ses clefs qu'avec des nœuds voisins, ce qui a pour effet d'augmenter le nombre de nœud qui doit être compromis pour que toutes les clefs du réseau deviennent accessibles. Les auteurs ont analysé les résultats d'une attaque "par complicité" sur la résiliance du réseau (le nombre de liens externes non compromis suite à la compromission d'un nombres de noeuds complices). Suite à cette analyse, ils ont proposé une heurestique qui migite l'impact de cette attaque grâce à une pré-distribution de clés qui favorisent le partage de clés entre noeuds voisins.

7 Evaluation des performances

Les auteurs de cet article ont effectué des simulations de leur protocole. Ils ont simulé des réseaux constitués de vingt à deux cents micros capteurs, et ils ont observé, dans un premier temps, comment le système se comportait pour différentes valeurs du nombre de clés administratives «k» et du nombre de clefs de cryptage «m», en prenant soin de choisir des valeurs k et m qui conservait la somme $(k + m)$ constante. Toutes les simulations ont été moyenné sur dix itérations avec des topologies différentes. Les résultats de la figure 3 montrent la relation entre différentes valeurs de k et m et le nombre correspondant de nœuds requis dans la chaîne de collusion pour compromettre toutes les cléfs du réseau.

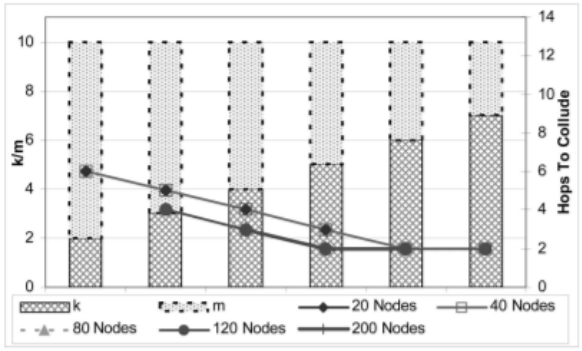

FIGURE 2

Les résultats obtenus montrent que le SHELL est toujours plus performant que l'assignation aléatoire de clefs et qu'il est même nettement plus performant pour les petites valeurs de k. Pour illustrer ce dernier point, considérons le cas k = 4, N = 120. Dans le cas du protocole SHELL il faut une longueur de chaîne de collusion de égale à 11 nœuds alors qu'il n'en faut que trois dans le cas d'une assignation de clefs aléatoire. Pour une probabilité d'une compromission de nœud "p" égale à 0,10 la probabilité de capturer un réseau qui utilise le protocole SHELL est inférieure d'un facteur 1016 à celle d'un réseau utilisant un protocole d'assignation aléatoire de clefs. Le résultat du graphique ci-dessus nous montre que la distribution aléatoire n'est pas avantageuse pour des petits réseaux pour la prévention de la collusion contrairement au SHELL. Pour ce qui est de l'énergie consommée, ils ont comparé les protocoles SHELL, Kerberos et l'approche de Jolly. On peut constater sur le schéma ci-dessous que le SHELL a une rentabilité bien meilleure que les deux autres. Dans un deuxième ensemble de simulations, les auteurs de l'article [1], on essayé d'évaluer le nombre de fois que où un adversaire pourrait capturer le réseau. Ils ont simulé des attaques par collusion sur un réseau constitué de cent micro capteurs avec k et m qui sont respectivement égaux à 3 et 7. Ils ont sélectionné au hasard un ensemble de noeuds compromis. Il ont effectué la simulation cent fois tout en faisait varier la taille de l'ensemble de noeuds compromis. Il ont effectué la même simulation sur la

méthode d'affection des clefs aléatoire [5]. A partir de la figure 3, nous pouvons voir que le nombre de fois que le réseau a été capturé dans le cadre de l'approche d'assignation de clé aléatoire est supérieur au nombre de fois qu'il a été capturé dans le cadre de l'utilisation du SHELL. Lorsque le nombre de noeuds compromis est peu élevé, SHELL augmente la résilience du réseau d'au moins un facteur de deux par rapport au cas de l'assignation aléatoire des clés. Toutefois, à mesure que de nouveaux noeuds sont compromis, le gain qu'offrait le SHELL commence à diminuer puisqu'il devient plus difficile de défendre le réseau sous une attaque de grande envergure. Plus globalement, les simulations menées et les résultats obtenus peuvent également être utiles pour déterminer les paramètres m et k pour le schéma EBS afin de trouver le meilleur compromis entre le stockage, la consommation d'énergie liée à la communication et la résilience.

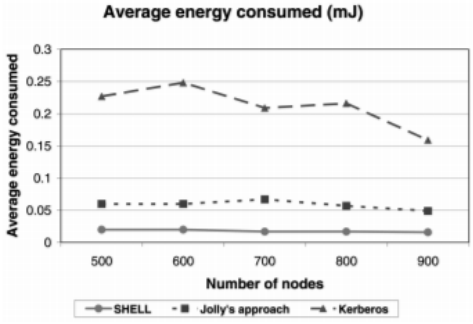

FIGURE 3

8 Critique

Tout d'abord je pense qu'il y a un problème au niveau de la comparaison énergétique. En dépit d'une comparaison bien argumentée il me semble que les auteurs de l'article auraient pu comparer leur protocole à davantage de protocoles ou le comparer au protocole réputé avoir le meilleur rendement énergétique. On peut faire la même réflexion en ce qui concerne l'amélioration des phénomènes de collusion, ils n'ont comparé leur protocole qu'au protocole d'assignation des clefs aléatoires alors qu'ils auraient pu comparer leurs résultats à d'autres protocoles basés eux aussi sur EBS. Outre le fait que ce protocole augmente de manière significative la résilience du réseau, il ne semble pas très dynamique et autonome car régulièrement il nécessite une intervention de la station de controle dans la gestion des clefs.

9 Conclusion

Le protocole Shell a pour but d'optimiser la pré-distribution des cléfs dans une topographie hiérarchqiue organisé en Cluster. Cette optimisation se base sur une construction particulière appelé EBS par cluster. Nous avons vu tout au long de cet article que le choix des paramètres de la construction EBS (k et m) est un critère essentiel. En effet, il faut trouver un compromis entre l'efficacité et la sécurité et ces critères peuvent être déterminés de manière optimale grâce aux expérimentations que les rédacteurs de l'article ont réalisées. Nous nous sommes aussi focalisé sur les améliorations qu'ils ont apporté à ce schéma comme une meilleur résistance aux attaques de type collusion.

10 References

[1] Mohamed F. Younis, Senior Member, IEEE, Kajaldeep Ghumman, and Mohamed Eltoweissy, «Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks», Aug.2006.
[2] Eltoweissy and al, «Combinatorial Optimization of Key Management in Group Communications» J. Network and Systems Management, vol. 12, no. 1, pp. 33-50, Mar. 2004.
[3] G. Gupta and M. Younis, "Load-Balanced Clustering in Wireless Sensor Networks," Proc. Int'l Conf. Comm. (ICC '03), May 2003.
[4] O. Younis and S. Fahmy, "HEED : A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," IEEE Trans. Mobile Computing, vol. 3, no. 4, pp. 366-379, Oct.-Dec. 2004.
[5] Laurent Eschenauer and Virgil D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks", Proceeding (CCS '02), pp. 41-47, Nov. 2002.