Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Đà Nẵng, ngày 17-18/08/2017
DOI: 10.15625/vap.2017.0006

CẢI TIẾN PHƯƠNG PHÁP PHÂN PHỐI KHOÁ MẬT MÃ AN TOÀN TRONG MẠNG CÓ TÀI NGUYÊN HẠN CHẾ

Nguyễn Đào Trường, Lê Mỹ Tú
Học viện Kỹ thuật mật mã, Hà Nội
truongnguyendao@gmail.com, tulemy@hotmail.com

TÓM TẮT: *Bài báo trình bày một phương pháp cải tiến lược đồ phân phối khóa an toàn trong mạng điều hành giám sát công nghiệp với mục tiêu chống lại những tấn công điển hình như tấn công phát lại, tấn công thông đồng nhau khi một thành viên bị trục xuất ra khỏi hệ thống với một thành viên mới gia nhập vào hệ thống hoặc hai/nhiều thành viên cùng gia nhập hệ thống. Ngày nay, mạng này do yêu cầu mà phải tiếp xúc rất nhiều với hệ thống mạng công cộng như Internet, WAN thì nguy cơ mất an toàn ngày càng hiện hữu. Những tấn công này có thể làm ảnh hưởng rất lớn đến hệ thống, nó có thể phá hủy toàn bộ hệ thống và để lại những hậu quả nghiêm trọng không chỉ đối với bản thân tổ chức sở hữu mạng mà còn ảnh hưởng tới an ninh, an toàn của quốc gia. Những cải tiến trong bài báo đảm bảo an toàn với những chi phí tính toán và lưu trữ tăng lên ít nhất.*

Từ khóa: *Cây hàm một chiều, kiến trúc phân cấp khóa logic, tấn công thông đồng, tấn công phát lại.*

I. GIỚI THIỆU

Mạng điều hành giám sát công nghiệp (ĐHGSCN) là một hệ thống mạng thực hiện việc điều khiển và giám sát trong các cơ sở hạ tầng quan trọng của quốc gia. Thông thường thì hệ thống mạng này thường là một mạng đóng ít có những giao tiếp với mạng công cộng bên ngoài. Ngày nay, do sự phát triển của công nghệ và đòi hỏi sự thích ứng với sự phát triển mạnh mẽ của Internet như IoT (Internet of Things) thì mạng ĐHGSCN ngày càng trở lên mất an toàn khi mà yêu cầu phải kết nối với những mạng mở bên ngoài. Vì vậy, vấn đề an toàn trong các hệ thống mạng ĐHGSCN càng trở lên cấp thiết. Để đảm bảo an toàn cho mạng này trong những môi trường công khai thì việc áp dụng các biện pháp bảo mật là một điều tất yếu. Theo tiêu chuẩn IEC 62351[1] trong lĩnh vực mạng công nghiệp đưa ra các yêu cầu phải áp dụng các giải pháp mã hóa/giải mã dữ liệu để đảm bảo an toàn trong quá trình truyền thông trong mạng. Tuy nhiên, vấn đề quan trọng nhất trong việc áp dụng mật mã lại nằm ở việc phân phối quản lý khóa phải đảm bảo an toàn, bí mật và hiệu quả.

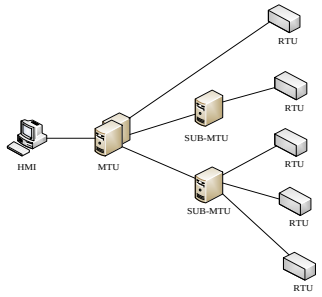
Thật đáng tiếc là những lược đồ quản lý khóa như SKE[2], SKMA[3] không hỗ trợ việc truyền quảng bá trong mạng, số lượng khóa lưu trữ lại mỗi nút mạng quá nhiều trong khi tài nguyên hạn chế và không có khả năng chống lại những tấn công phát lại (đây là tấn công rất nguy hiểm trong mạng ĐHGSCN). Để giải quyết vấn đề này, một giải pháp hiệu quả sử dụng mô hình LKH (Logical Key Hierachy) [6], [7] kết hợp với OFT (One-way Function Tree) [4], [5]. Tổng chi phí truyền thông của việc quản lý nhóm khi một thành viên ra khỏi nhóm giảm xuống còn $\log_2 n$, với n là tổng số người dùng trong hệ thống, bằng 1/2 chi phí truyền thông trong kiến trúc LKH. Tuy nhiên, Horng [8] đã phát hiện OFT có lỗ hổng tấn công dạng thông đồng. Tấn công này là một thảm họa lớn như trong chăm sóc sức khỏe và giám sát bệnh nhân [9] [10], và kiến trúc thu thập dữ liệu môi trường [11] [12] [13] [14], đặc biệt với việc phát hiện những sự kiện nóng bỏng [15] [16].

Trong bài báo này, chúng tôi đề xuất lược đồ với tên OFT-1 là lược đồ cải tiến từ lược đồ OFT để chống lại các tấn công thông đồng và tấn công phát lại. Lược đồ đề xuất này hiệu quả trong việc giải quyết bài toán an toàn với chi phí tăng tối thiểu trong tính toán và truyền thông.

II. MẠNG ĐIỀU HÀNH GIÁM SÁT CÔNG NGHIỆP

A. Cấu trúc mạng ĐHGSCN

Hệ thống mạng ĐHGSCN thực hiện giám sát và điều khiển các cơ sở từ xa thông qua việc thu thập dữ liệu từ các bộ cảm biến khác nhau trong mạng ĐHGSCN. Hệ thống mạng ĐHGSCN thường có cấu trúc phân cấp. Những dạng truyền thông của hệ thống này cũng là cấu trúc master-slave. Hình 1 mô tả cấu trúc đơn giản của hệ thống ĐHGSCN.



Hình 1. Cấu trúc phân cấp của hệ thống ĐHGSCN

Hệ thống ĐHGSCN thông thường có ba loại thiết bị truyền thông gồm HMI (Human Machine Interface), MTU (Master Terminal Unit) hoặc/và SUB-MTU và RTU (Remote Terminal Unit). Kiến trúc mạng của hệ thống ĐHGSCN thường là tĩnh ít có những biến động. Các đường truyền giữa các nút được biết trước, chỉ có một vài thay đổi trong mạng khi thêm hoặc bớt RTU hoặc SUB-MTU. Quá trình truyền thông chỉ xuất hiện giữa HMI với MTU, MTU với SUB-MTU, hai SUB-MTU với nhau, MTU với RTU, hai RTU với nhau. Truyền thông HMI-MTU có thể được thực hiện dễ dàng qua dịch vụ web sử dụng các giao thức cơ bản trong bộ giao thức TCP/IP. Tuy nhiên, truyền thông HMI-MTU ít có những giới hạn về tài nguyên hơn so với các truyền thông còn lại.

B. Những ràng buộc và yêu cầu hệ thống

Mạng ĐHGSCN khác với các môi trường mạng thông thường do môi trường hoạt động của nó thường nằm trong các cơ sở hạ tầng quan trọng của quốc gia. Vì vậy, mạng này có một số ràng buộc sau:

- Khả năng tính toán hạn chế: Những thiết bị ở xa như các RTU là một hệ thống nhúng có không gian lưu trữ và khả năng tính toán thấp.
- Tốc độ truyền dữ liệu thấp: Vì hệ thống ĐHGSCN được sử dụng trong thời gian dài, đường truyền trong mạng có băng thông thấp.
- Xử lý thời gian thực: Hệ thống ĐHGSCN cần chính xác. Độ trễ trong quá trình xử lý dữ liệu có thể dẫn đến một số vấn đề nguy hiểm.

Những ràng buộc của hệ thống ĐHGSCN ở trên làm cho nó khó được áp dụng những công nghệ an toàn đòi hỏi tính toán lớn, vì vậy những ràng buộc đó được xem là cơ sở cho việc áp dụng những cơ chế an toàn.

C. Những ràng buộc về an toàn

Trong mạng ĐHGSCN thì những nguy cơ phải đối mặt với vấn đề mất an toàn đó là tấn công phát lại và tấn công thông đồng. Ngoài ra, trong lược đồ quản lý và phân phối khóa còn phải đảm bảo: *Bảo toàn bí mật trước* (Các thành viên đã bị trục xuất ra khỏi hệ thống không được truy cập vào những thông tin mới trong hệ thống, ở trạng thái đó chúng không thể tính (hoặc truy cập) những khóa mới sau này) và *Bảo toàn bí mật sau* (những thành viên mới vào hệ thống không thể truy cập những thông tin trước khi anh ta gia nhập, ở trạng thái này chúng không thể tính ra những khóa cũ). Do đó, khi xây dựng một hệ thống an toàn thì ngay trong việc quản lý và phân phối khóa cũng cần phải tính đến những điều kiện này. Vì vậy, một hệ thống quản lý khóa an toàn và hiệu quả cần phải đảm bảo:

- Chống tấn công phát lại;
- Chống tấn công thông đồng;
- Bảo toàn bí mật trước;
- Bảo toàn bí mật sau.

III. LƯỢC ĐỒ QUẢN LÝ KHÓA OFT VÀ TẤN CÔNG THÔNG ĐỒNG

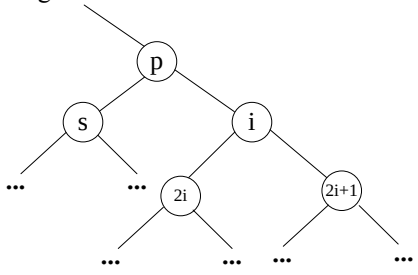
A. Lược đồ quản lý khóa OFT

OFT là một lược đồ quản lý khóa được Sherman và cộng sự đề xuất trong [4], [5]. Nó dựa trên lược đồ LKH [6], [7] và sử dụng hàm một chiều trong quản lý khóa [17]. Trong OFT, có một người quản lý nhóm tập trung thực hiện các việc cập nhật khóa, lưu trữ khóa và phân phối khóa. Cấu trúc quản lý của OFT là một cây khóa nhị phân. Trong cây này, mỗi nút i là sự kết hợp một bí mật nút x_i, một bí mật nút mù y_i và một khóa nút K_i.

1. Định nghĩa 1

Bí mật nút mù là kết quả đầu ra của hàm một chiều với đầu vào là khóa của nút đó, bí mật nút mù được sử dụng để tìm các khóa ở nút cao hơn trong cây khóa.

Bí mật nút của nút gốc của cây chính là khóa nhóm. Bí mật nút mù y_i được tính theo công thức $y_i=f(x_i)$, khóa nút K_i được tính theo công thức $K_i=g(x_i)$, với f và g là các hàm một chiều chuyên dụng khác nhau. K_i được sử dụng để mã hóa thông tin khóa cập nhật khi thu hồi khóa. Mỗi thành viên trong nhóm lưu trữ các bí mật nút mù của anh em của các nút đó theo đường dẫn từ nút đó đến nút gốc. Do đó, mỗi thành viên có thể sử dụng bí mật nút lá của nó và các bí mật nút mù để tính các khóa nút khác theo đường dẫn từ dưới lên.



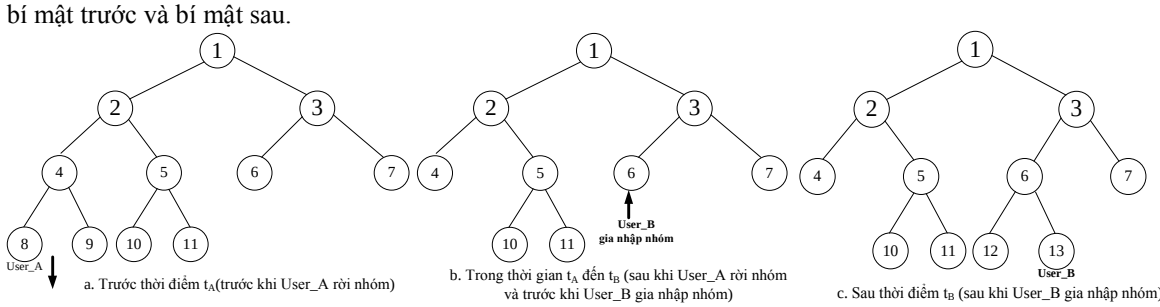
Hình 2. Cấu trúc cây khóa hàm một chiều

Trong Hình 2, *i* là một nút trong cây khóa. Con trái và con phải tương ứng là *2i* và *2i +1*. Những người dùng trong cây con có gốc tại *i* có thể tính bí mật nút *i* theo công thức $x_i = y_{2i} \oplus y_{2i+1} = f(x_{2i}) \oplus f(x_{2i+1})$, với $\oplus$ là phép XOR bít. Chúng cũng có thể tính bí mật nút mù y_i theo công thức $y_i=f(x_i)$. Những người dùng này lưu trữ bí mật nút mù y_s kết hợp với nút *s* là anh em của nút *i*. Vì vậy, chúng có thể tính bí mật nút cha *p* theo công thức $x_p = y_s \oplus y_i$. Tương tự, mỗi thành viên có thể tính các bí mật nút trên đường dẫn từ nút đó đến nút gốc, kể cả bí mật nút gốc (khóa nhóm).

Khi một bí mật nút trong cây khóa thay đổi thì bí mật nút mù mới được gửi đến tất cả người dùng lưu trữ bí mật nút mù cũ trong nhóm để cập nhật lại khóa. Chẳng hạn, giả sử bí mật nút *i*, x_i thay đổi. Bí mật nút mù mới y_i được gửi đến những người dùng lưu trữ bí mật mù cũ của nút *i*. Những người dùng này chỉ là con của *s*. Con của *s* tính khóa nút K_s, người quản lý nhóm chỉ cần mã hóa bí mật nút mù mới y_i bằng khóa K_s. Sau đó, người quản lý nhóm gửi quảng bá thông tin khóa đã mã hóa đó đến nhóm. Như thế, bí mật nút mù mới được gửi đến toàn bộ thành viên của nhóm.

B. Tấn công thông đồng trong OFT

Trong [8], Horng đã chỉ ra lỗ hổng các tấn công thông đồng trong OFT. Sau đó, ông ta kết luận rằng OFT không bảo toàn bí mật trước và bí mật sau.



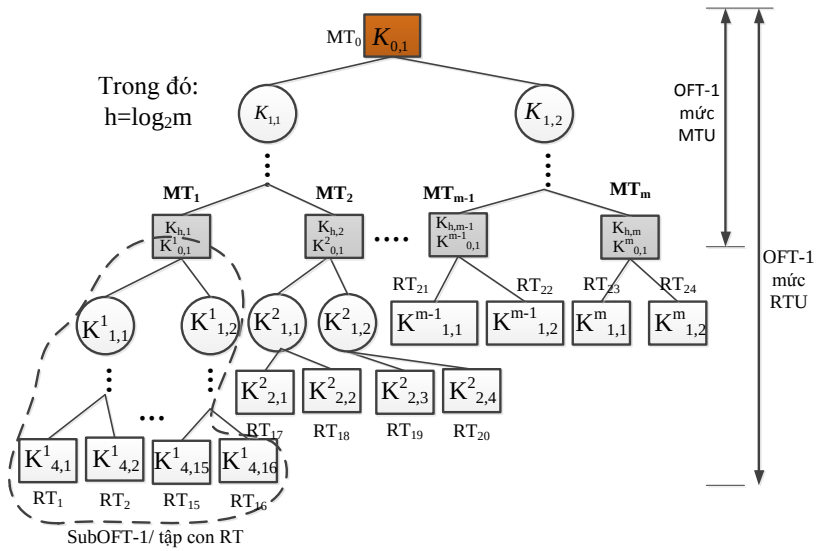
Hình 3. Tấn công thông đồng trong OFT

Hình 3 mô tả những thay đổi các thành viên trong nhóm. Nhìn vào Hình 3, đầu tiên User_A (ở nút 8) rời khỏi nhóm tại thời điểm t_A. Sau đó, User_B gia nhập nhóm tại thời điểm t_B. Hình 3 mô tả các cây khóa trước khi User_A rời khỏi nhóm trong khoảng t_A đến t_B, sau đó User_B gia nhập nhóm. Trong thời gian từ t_A đến t_B không có bất kỳ người dùng nào gia nhập nhóm hay rời khỏi nhóm. Ký hiệu $x_i[t_A,t_B]$ là bí mật nút *i*, $y_i[t_A,t_B]$ là bí mật nút mù của nút *i* trong khoảng từ t_A đến t_B. Sau khi User_A rời khỏi nhóm, x_3 không bị thay đổi cho đến khi User_B gia nhập. Vì vậy, User_A nắm giữ bí mật mù của nó là $y_3[t_A,t_B]$. x_2 bị thay đổi khi User_A rời khỏi nhóm và những bí mật còn lại không bị thay đổi cho đến khi User_B gia nhập nhóm. Khi User_B gia nhập nhóm, anh ta nhận được bí mật nút mù của nó là $y_2[t_A,t_B]$. Tựu chung lại, chúng biết $y_2[t_A,t_B]$ và $y_3[t_A,t_B]$. Vì vậy, chúng có thể thông đồng với nhau để tính ra khóa nhóm trong khoảng thời gian $[t_A, t_B]$ theo công thức $x_1[t_A,t_B] = y_2[t_A,t_B] \oplus y_3[t_A,t_B]$.

User_A tính được khóa nhóm mới sau khi rời khỏi nhóm, vì vậy OFT lỗi trong việc bảo toàn bí mật trước. User_B tính khóa nhóm trước khi anh ta gia nhập nhóm, vì vậy OFT lỗi trong việc bảo toàn bí mật sau.

IV. LƯỢC ĐỒ QUẢN LÝ KHÓA ĐỀ XUẤT (OFT-1)

Trong bài báo này, chúng tôi đề xuất một giao thức quản lý và phân phối khóa an toàn và hiệu quả trong việc đảm bảo truyền thông an toàn trong mạng ĐHGSCN.



Hình 4.Cấu trúc của OFT-1

A. Một số định nghĩa và ký hiệu

Trong bài báo này chúng tôi sử dụng một số định nghĩa và ký hiệu trong Bảng 1.

Bảng 1. Các định nghĩa và ký hiệu

Ký hiệu	*Ý nghĩa*	*Ký hiệu*	*Ý nghĩa*
h	Chiều cao của cây	$\min(i, j)$	Số nhỏ nhất giữa *i* và *j*.
m	Số SUB-MTU	K_i	Khóa bí mật của nút *i*
n	Số RTU tối đa mà một SUB-MTU quản lý	*TVP*	Kết hợp của đánh dấu thời gian và số thứ tự.
N_{MT_i}	Số RTU mà SUB-MTU thứ i quản lý tại nút MT_i	$E_K(D)$	Hàm mã hóa AES với khóa bí mật *K*
KDC	Trung tâm quản lý và phân phối khóa tại một MTU	$H(D)$	Hàm băm
MT_i	Nút SUB-MTU thứ i trong nhóm SUB-MTU hiện tại $(i \geq 1)$	A→B:{Msg}	A gửi thông điệp Msg cho B. A và B có thể là một tập các thực thể. Tập các thực thể được biểu diễn {*các thực thể*}
MT_0	Nút MTU	$SK_{i,j}$	Khóa phiên giữa nút *i* và nút *j*
$K_{i,j}$	Khóa thứ j tại mức i trong cây nhị phân tương ứng với tập MT, $K_{0,1}$ là khóa nút gốc của cây nhị phân. $0 \leq i \leq h, 1 \leq j \leq m$	$K_{i,j}^s$	Khóa thứ j tại mức i trong cây nhị phân tương ứng với tập con RT của MT_s. $0 \leq s \leq m$, $0 \leq i \leq \log_2 n$, $1 \leq j \leq m$
RT	Tập các thành viên RTU hiện tại, $RT = \{RT_1, RT_2, ..., RT_{mn}\}$	MT	Tập các thành viên SUB-MTU hiện tại, $MT = \{MT_1, MT_2, ..., MT_m\}$

B. Khởi tạo hệ thống

KDC xây dựng cấu trúc khóa bằng cách sử dụng cấu trúc LKH như trong Hình 4. Cấu trúc này gồm hai tập, *MT* và *RT*. Cấu trúc khóa cho mỗi tập này được xây dựng theo LKH. MTU đóng vai trò quản lý chung cả nhóm, khóa nhóm tại đây được sử dụng để truyền giữa MTU với các SUB-MTU. SUB-MTU đóng vai trò là quản lý nhóm con, khóa ở đây gọi là khóa nhóm con được sử dụng để truyền thông giữa các RTU với MTU hoặc các RTU với các SUB-MTU còn lại trong nhóm mà MTU quản lý.

1. Cấu trúc khóa của tập MT

Các khóa trong *MT* được tổ chức theo cấu trúc cây nhị phân gồm 2*m*-1 nút, với chiều cao cây là $h = \log_2 m$ với *m* là số SUB-MTU. Trong cấu trúc khóa này, các khóa tại các nút lá $MT_i (1 \leq i \leq m)$ là $K_{h,j} (1 \leq j \leq m)$ được gán cho tất cả các SUB-MTU. Còn các khóa khác $K_{i,j} (1 \leq i \leq h-1, 1 \leq j \leq m)$ được tính theo biểu thức (1). Nói cách khác, khóa được tạo ra bằng cách sử dụng hàm băm với đầu vào là các giá trị băm của các nút con của nó.

$$K_{i+1,\lceil j/2 \rceil} = H\left(H\left(K_{i,j}\right), H\left(K_{i,j+1}\right)\right) \text{ với } (1 \leq i \leq h-1, 1 \leq j \leq m) \quad (1)$$

Khóa của nút gốc MT_0 là $K_{0,1}$ được tạo ra từ biểu thức (1). Do đó, MTU biết khóa nút gốc từ các khóa $K_{h,j}$ $(1 \leq j \leq m)$ của *m* nút SUB-MTU.

2. Cấu trúc của tập RT

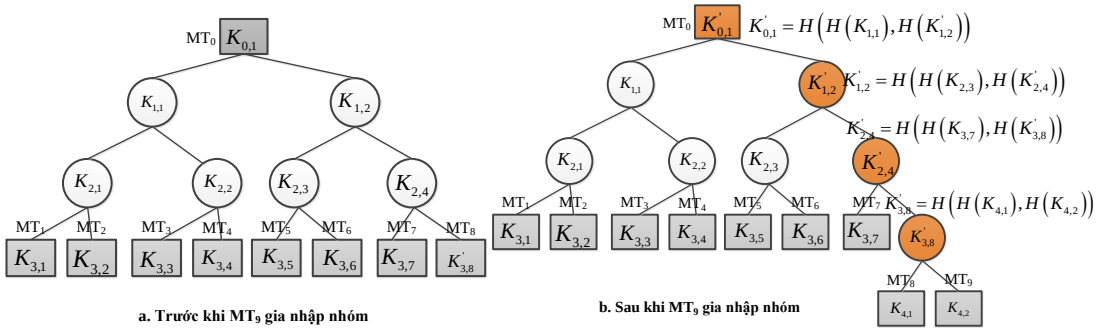
Mỗi tập *RT* cũng được tổ chức theo một cây nhị phân gồm $2\left(N_{MT_i}\right) - 1$ nút, do đó chiều cao của cây là $h = \log_2\left(N_{MT_i}\right)$. Trong cấu trúc khóa của tập con *RT* do MT_i quản lý thì các khóa của các nút lá $RT_j (1 \leq j \leq n)$ là $K^i_{\log_2\left(N_{MT_i}\right), j}$ và được gán cho toàn bộ các RTU. Các khóa còn lại $K^s_{i,j} (1 \leq s \leq m, 1 \leq j \leq n, 1 \leq i \leq \log_2 n)$ được tạo ra bằng cách sử dụng hàm băm với đầu vào là các giá trị băm từ khóa của các nút con của nó theo biểu thức (2).

$$K^s_{i+1,\lceil j/2 \rceil j} = H\left(H\left(K_{i,j}\right), H\left(K_{i,j+1}\right)\right) \text{với } \left(1 \leq i \leq \log_2\left(N_{MT_s}\right) - 1, 1 \leq j \leq n\right) \quad (2)$$

Khóa nút gốc $K_{0,1}^{i}$ của tập con *RT* do MT_i quản lý cũng được tạo ra từ biểu thức (2). Do đó, SUB-MTU tương ứng với nút MT_i biết khóa nút gốc từ các khóa $K^{i}_{\log_2(N_{MT_i}),j}\ (1 \le j \le n)$ của *n* RTU.

C. Thêm một RTU hoặc SUB-MTU vào hệ thống

Đầu tiên KDC phải tìm một nút gần nút gốc nhất để bổ sung thêm nút mới này. Tiếp theo, nút tại vị trí sẽ được bổ sung đó sẽ trở thành nút con trái còn nút mới sẽ trở thành nút con phải. KDC tạo một khóa mới để cấp cho nút mới này và nút mới chuyển xuống đó. Để đảm bảo bảo toàn bí mật sau trong quá trình gia nhập nhóm, toàn bộ các khóa mà một RTU hoặc SUB-MTU mới gia nhập nhận được thì KDC phải cập nhật lại toàn bộ các khóa của các nút từ nút mới đến nút gốc và các nút anh em của các nút trên đường dẫn đó. Do đó, mỗi khi thực hiện việc bổ sung thành viên mới, thì toàn bộ các khóa của các nút mà nút mới vào có thể biết đã được cập nhật lại bằng cách sử dụng hàm băm để cập nhật lại khóa. Điều này đảm bảo chống tấn công thông đồng của người mới gia nhập nhóm, đồng thời đảm bảo bảo toàn bí mật sau.



Hình 5. Bổ sung một thành viên mới trong OFT-1

1. Thêm một SUB-MTU mới

Khi bổ sung một SUB-MTU mới vào hệ thống, cấu trúc khóa tại tập *MT* thay đổi và khóa nhóm mới được tạo ra cho toàn bộ SUB-MTU được phân phối đến toàn bộ các SUB-MTU theo thuật toán 1. Hình 5 mô tả chi tiết quá trình bổ sung thêm một SUB-MTU vào hệ thống.

Thuật toán 1: AddnewSUB-MTU()

Input: OFT *T*, SUB-MTU mới MT_{m+1}

Output: *T* cập nhật.

1. *KDC* chọn một nút lá phù hợp *x* = SelectLeaftoAdd(*T*);
2. OldMember = Member(*x*); (*a*, *b*) = Split(*x*); *a* = OldMember; *b* = MT_{m+1}
3. *KDC* tạo ngẫu nhiên một khóa mới $K_{MT_{m+1}}$ và gán cho MT_{m+1}
4. *KDC* tính lại K_{MT_m} (K'_{MT_m}) từ hai khóa mới $K_{MT_{m+1}}$, MT_{m+1} của hai nút lá (trong hình 5 là $K'_{3,8}$)
5. *KDC* mã hóa khóa mới K'_{MT_m} bằng khóa cũ K_{MT_m} rồi gửi unicast cho MT_m: $KDC \xrightarrow{unicast} MT_m : \left\{E_{K_{MT_m}}(K'_{MT_m})\right\}$
6. *KDC* cập nhật lại toàn bộ các khóa trên đường dẫn từ nút mới gia nhập đến nút gốc bằng cách sử dụng các hàm băm với đầu vào lần lượt là các giá trị băm của khóa của các nút con. Trong hình 5 ở đây gồm: $K'_{3,8} = H\left(H\left(K_{4,1}\right), H\left(K_{4,2}\right)\right)$;

   $K'_{2,4} = H\left(H\left(K_{3,7}\right), H\left(K'_{3,8}\right)\right)$; $K'_{1,2} = H\left(H\left(K_{2,3}\right), H\left(K'_{2,4}\right)\right)$; $K'_{0,1} = H\left(H\left(K_{1,1}\right), H\left(K'_{1,2}\right)\right)$.
7. *KDC* mã hóa các khóa mới này bằng các khóa cũ rồi truyền multicasts các khóa cập nhật này đến các thành viên tương ứng, và truyền unicasts đến nút mới.

   $$KDC \xrightarrow{multicast} \{MT\} : \left\{E_{K_{i,j}}(K'_{i,j},...)\right\}$$

   $$KDC \xrightarrow{unicast} MT_{m+1} : \left\{E_{K_{MT_{m+1}}}(K'_{i,j},...)\right\}$$
8. *KDC* gửi thông điệp nhắc tất cả các thành viên là anh em của các nút trên đường dẫn từ nút mới đến nút gốc là có thành viên mới và cập nhật lại toàn bộ các khóa mà chúng nắm giữ.
9. KDC truyền unicast khóa nút cha của nút mới đến nút anh em của nút mới. Trong hình 5 là $KDC \xrightarrow{unicast} \{MT_8\} : \left\{E_{K_{MT_8}}(K'_{0,1}, K'_{1,2}, K'_{2,4}, K'_{3,8})\right\}$.

2. Thêm RTU mới

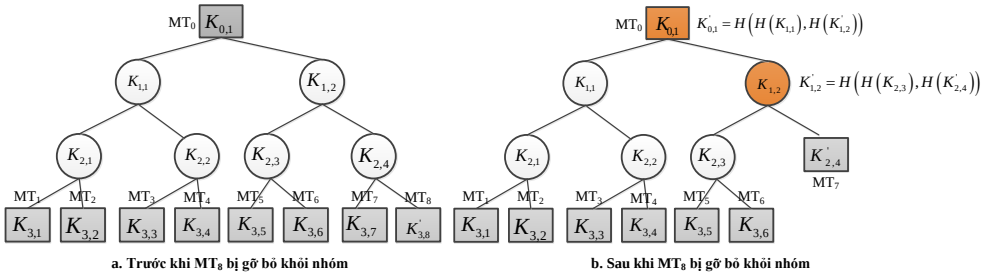
Khi bổ sung thêm một RTU mới cũng được thực hiện tương tự như bổ sung SUB-MTU nhưng ở tầng RTU và các khóa cập nhật phải lên đến MT_0.

D. Hủy một RTU hoặc SUB-MTU ra khỏi hệ thống

Khi gỡ một RTU hoặc SUB-MTU ra khỏi hệ thống thì cũng cần phải cập nhật lại toàn bộ các khóa liên quan tới RTU hoặc SUB-MTU bị gỡ bỏ đó. Quá trình hủy một SUB-MTU (hoặc RTU) được mô tả như trong Hình 6.

1. Hủy một SUB-MTU

Khi hủy một SUB-MTU ra khỏi hệ thống, cấu trúc khóa của tập MT thay đổi và khóa nhóm mới chỉ là với những SUB-MTU còn lại và khóa này phải được tính lại và phân phối đến các SUB-MTU theo thuật toán 2.



Hình 6. Hủy bỏ một thành viên trong OFT-1

Thuật toán 2: RemoveSUB-MTU()

Input: OFT *T*, SUB-MTU bị hủy MT_m

Output: *T* cập nhật.

1. y= *leaf*(MT_k); p = Parent (y); s=sibling(y);
2. if Leaf(s) = Yes then Member(s) = p; $x = s$;
   Else $p = s$; $x =$ SelectLeaf(s);
3. *KDC* tạo một khóa mới $K'_{MT_{m-1}}$ rồi cấp cho MT_{m-1}, trong hình 6 là $K'_{2,4}$ được cấp mới cho MT_7.
4. *KDC* cập nhật lại toàn bộ các khóa trên đường dẫn từ nút bị hủy bỏ đến nút gốc bằng cách sử dụng các hàm băm với đầu vào lần lượt là các giá trị băm của khóa của các nút con. Trong hình 6 ở đây là: $K'_{1,2} = H\left(H\left(K_{2,3}\right), H\left(K'_{2,4}\right)\right)$;
   $K'_{0,1} = H\left(H\left(K_{1,1}\right), H\left(K'_{1,2}\right)\right)$.
5. *KDC* mã hóa các khóa mới này bằng các khóa cũ rồi truyền multicasts các khóa cập nhật này đến các thành viên tương ứng, và truyền unicast đến nút mới chuyển lên nút cha.
$$KDC \xrightarrow{multicast} \{MT\} : \left\{E_{K_{i,j}}(K'_{i,j},...)\right\}$$
$$KDC \xrightarrow{unicast} MT_{m-1} : \left\{E_{K_{MT_{m-1}}}(K'_{i,j},...)\right\}$$
6. *KDC* gửi thông điệp nhắc tất cả các thành viên là anh em của các nút trên đường dẫn từ nút mới đến nút gốc là có thành viên vừa bị hủy và cập nhật lại toàn bộ các khóa mà chúng nắm giữ.
7. KDC truyền unicast khóa nút cha của nút mới đến nút anh em của nút mới. Trong hình 6 là $KDC \xrightarrow{unicast} \{MT_7\} : \left\{E_{K_{MT_7}}(K'_{0,1}, K'_{1,2})\right\}$.

2. Hủy một RTU

Quá trình gỡ một RTU ra khỏi hệ thống cũng được thực hiện tương tự như hủy một SUB-MTU nhưng ở mức RTU và việc cập nhật lại khóa được hiệu chỉnh lên đến tận MT_0.

E. Tính khóa phiên khi truyền dữ liệu

Quá trình truyền dữ liệu có thể xảy ra một trong hai trường hợp:

1. Node tới Node

Quá trình truyền node tới node gồm MTU đến SUB-MTU, RTU đến RTU, SUB-MTU đến SUB-MTU, SUB-MTU đến RTU. Khóa phiên để một nút i truyền cho nút j được tính theo công thức $SK_{i,j} = H\left(K_{u,v}, ID_i, ID_j, TVP\right)$. Trong đó, $K_{u,v}$ là khóa dùng chung của nút j và nút i thông qua cấu trúc OFT-1.

2. Node tới nhóm

Quá trình truyền node tới nhóm gồm MTU đến các SUB-MTU, SUB-MTU đến các RTU, MTU đến các RTU. Khóa phiên để một nút i truyền cho các thành viên trong nhóm j được tính theo công thức $SK_{i,j} = H\left(K_{u,v}, TVP\right)$. Trong đó, $K_{u,v}$ là khóa dùng chung của tất cả các thành viên trong nhóm j và nút i. Thông qua cấu trúc OFT-1 thì nhiều nút có thể sử dụng cùng một khóa.

F. So sánh với các lược đồ khác

Trong bài báo này, chúng tôi sử dụng AES-128 làm thuật toán mã hóa, SHA-1 làm hàm băm. Chúng tôi tập trung vào so sánh về mặt lý thuyết, đặc biệt là độ phức tạp tính toán hay chi phí tính toán trong các vấn đề tổng chi phí truyền thông, chi phí tính toán của KDC, chi phí tính toán tối đa của các thành viên và tổng chi phí lưu trữ tối đa của các thành viên. Ngoài ra bài báo cũng đánh giá về một số vấn đề an toàn như chống tấn công thông đồng, chống tấn công phát lại. Bảng 2 đưa ra so sánh chi tiết của lược đồ đề xuất với các lược đồ trước đó. Trong đó AC là tấn công thông đồng, RA là tấn công phát lại, h là chiều cao của cây khóa, L kích thước khóa, C_E là chi phí tính toán mã hóa, C_D là chi phí tính toán giải mã, C_H là chi phí tính toán hàm băm, C_f là chi phí tính toán hàm một chiều cửa sập, C_M là chi phí tính toán phép nhân Modulo.

Trong OFT-1 giả sử có n người dùng, tổng chi phí truyền thông của quản lý hệ thống giống như trong OFT. Điều này chủ yếu là do không mở rộng thêm những thông tin khóa cần được gửi đi trong nhóm khi thực hiện thu hồi khóa. Khi một người dùng mới gia nhập hệ thống, để ngăn tấn công thông đồng thì những người dùng trong hệ thống chỉ cần thực hiện các hàm một chiều. So với OFT thì trong OFT-1 người quản lý nhóm chịu chi phí tính toán thêm $\log_2 n$ -1 bí mật nút mù. Các thành viên trong hệ thống chịu tổng chi phí lưu trữ thêm ít nhất là $\log_2 n$-1 bí mật nút mù. Vì $C_h < C_f$ nên chi phí tính toán của người quản lý nhóm và chi phí tính toán tối đa của các thành viên thấp hơn trong HOFT[20].

Từ những phân tích trên cho thấy so với lược đồ OFT thì lược đồ đề xuất chỉ chịu tăng thêm nhỏ về chi phí tính toán của người quản lý nhóm và tổng chi phí lưu trữ của các thành viên. Tổng chi phí truyền thông của người quản lý nhóm giống như OFT. Quan trọng hơn là tổng chi phí truyền thông và chi phí tính toán của lược đồ đề xuất thấp hơn so với các lược đồ khác.

Bảng 2. So sánh lược đồ đề xuất với các lược đồ khác

Lược đồ	*Chống AC*	*Chống RA*	*Tổng chi phí truyền thông của KDC (thêm;hủy)*	*Chi phí tính toán của KDC (thêm;hủy)*	*Tổng chi phí tính toán của các thành viên (thêm;hủy)*
OFT-1	**Có**	**Có**	$(2h+1)*L$; $(h+1)*L$	$(2h+1)*C_E+(2h-1)*C_H$; $(h+1)*C_E+(h-1)*C_H$	$2C_D+(2h-1)*C_H$; C_D+2h*C_H
OFT [4][5]	Không	Không	$(2h+1)*L$; $(h+1)*L$	$(2h+1)*C_E+(h-1)*C_H$; $(h+1)*C_E+h*C_H$	$2C_D+h*C_H$; C_D+h*C_H
Ku và cộng sự [18]	Có	Không	$(2h+1)*L$; $(h+1)*L$	$(2h+1)*C_E+(h-1)*C_H$; $(h^2+h+1)*C_E+(h^2+h)*C_H$	$2C_D+h*C_H$; $h*C_D+(1/2)h^2*C_H$
Xu và cộng sự [19]	Có	Không	$(2h+1)*L$; $(h+1)*L$	$(2h+1)*C_E+(h-1)*C_H$; $(h+1)*C_E+(h-2)*C_H$	$2C_D+h*C_H$; C_D+h*C_H
HOFT [20]	Có	Không	$(2h+1)*L$; $(h+1)*L$	$(2h+1)*C_E+(2S*h+1)*C_M+$ $(h+S*h+1)*C_f$; $(h+1)*C_E+(h+2)*C_M+(h-1)C_f$	$(1+h)*C_H+2h*C_M+$ $(h+S*h)*C_f$; $C_D+(h+1)*C_M+h*C_f$
SKE/ SKMA [2][3]	Không	Không	-	-	-

V. CHỨNG MINH AN TOÀN CỦA LƯỢC ĐỒ ĐỀ XUẤT

Để chứng minh lược đồ đề xuất OFT-1 của chúng tôi an toàn. Chúng tôi sử dụng mô hình an toàn của Panjwani phát triển trong [21]. Ở đây chúng tôi sử dụng khái niệm người dùng thay cho RTU, SUB-MTU, người dùng ở HMI. Xét một hệ thống gồm n người dùng, được gán nhãn từ 1, 2, ..., n. Mỗi người dùng i dùng chung khóa bí mật K_i với quản lý nhóm. Tại thời điểm t, những người dùng trong tập $S^{(t)} \subseteq \{1, 2, ..., n\}$gọi là những người dùng trong nhóm tại thời điểm t đó. Ký hiệu $T^{(t)}$ là cây OFT-1 tương ứng với tập $S^{(t)}$. Ký hiệu $[n]$ là tập $\{1, 2, ..., n\}$ và $2^{[n]}$ là tập lũy thừa của $[n]$. Ta có $\overrightarrow{S^{(t)}}=(S^{(0)}, S^{(1)}, ..., S^{(t)}) \in (2^{[n]})^t$. Nếu với mọi $t \geq 1$, $S^{(t-1)}$ thay thành $S^{(t)}$ qua phép thay đổi đơn, thì dãy $\overrightarrow{S^{(t)}}$ gọi là dãy đơn. Theo đó, ta có một số định nghĩa và bổ đề sau:

A. Một số định nghĩa

1. Định nghĩa 2

Giao thức OFT-1 gồm n người dùng gọi là đúng nếu với $\forall t \geq 0$, dãy đơn $\overrightarrow{S^{(t)}}$, $\forall i \in S^{(t)}$ thì i chỉ biết các khóa bí mật nút trong đường dẫn từ nút lá của nó đến nút gốc và các bí mật nút mù của nút anh em trên đường dẫn này và không biết các khóa bí mật hoặc bí mật nút mù khác trong $T^{(t)}$.

2. Định nghĩa 3

Giao thức OFT-1 gồm *n*-người dùng được gọi là an toàn chống tấn công người dùng đơn nếu với $\forall t \geq 0$, dãy đơn $\overrightarrow{S^{(t)}}$, $\forall i \notin S^{(t)}$ thì *i* không bao giờ có thể khôi phục lại bất kỳ khóa bí mật nút nào trong $T^{(t)}$ từ K_i và những thông điệp thu hồi khóa.

3. Định nghĩa 4

Giao thức OFT-1 gồm *n*-người dùng được gọi là an toàn chống các tấn công thông đồng nếu với $\forall t \geq 0$, dãy đơn $\overrightarrow{S^{(t)}}$, tập người dùng bất kỳ $U = \{i | i \notin S^{(t)}\}$, *U* không thể khôi phục lại bất kỳ khóa bí mật nào trong $T^{(t)}$ từ $\{K_i | i \in U\}$ và những thông điệp thu hồi khóa.

B. Một số bổ đề

1. Bổ đề 1

Giao thức OFT-1 đúng và an toàn chống tấn công người dùng đơn.

Chứng minh

Với $t = 0$, thì $S^{(0)} = \phi$, phát biểu đó đúng. Chúng ta sẽ chứng minh nếu phát biểu đó đúng với $t \geq 0$ thì nó cũng đúng với $t+1$. Với chuỗi đơn $\overrightarrow{S^{(t+1)}} = (S^{(1)}, S^{(2)}, ..., S^{(t+1)}) \in (2^{[n]})^{t+1}$, xét các trường hợp sau:

a) Trường hợp 1:

Với $(i \in S^{(t)} \wedge i \in S^{(t+1)})$, trong OFT-1, *i* chỉ có thể khôi phục lại những khóa bí mật nút và bí mật nút mù đã thu hồi khóa mà nó nắm giữ trong $T^{(t)}$ khi cần từ thông điệp thu hồi khóa. Do đó, nó chỉ nắm giữ tất cả những khóa bí mật nút và bí mật nút mù trong $T^{(t+1)}$ khi cần.

b) Trường hợp 2:

Với $(i \notin S^{(t)} \wedge i \in S^{(t+1)})$, điều đó có nghĩa là *i* gia nhập tại thời điểm *t*. Trong OFT-1, thành viên gia nhập mới *i* chỉ khôi phục được những khóa bí mật nút và bí mật nút mù cần thiết từ những thông điệp thu hồi khóa.

c) Trường hợp 3:

Với $(i \in S^{(t)} \wedge i \notin S^{(t+1)})$, điều đó có nghĩa là *i* bị hủy (rời khỏi hệ thống) tại thời điểm $t+1$. Từ giả thuyết, *i* chỉ biết những khóa bí mật nút trên đường dẫn của nó đến nút gốc và những bí mật nút mù của anh em trên đường dẫn này trong $T^{(t)}$. Trong OFT-1, toàn bộ bí mật nút trong đường dẫn của *i* đến các gốc trong $T^{(t)}$được thay đổi tại thời điểm $t+1$. Mặc dù một số bí mật nút mù trong $T^{(t)}$ mà *i* đưa ra có thể không thay đổi tại thời điểm $t+1$, các thông điệp thu hồi khóa được mã hóa bằng các khóa bí mật nút anh em của nó theo đường dẫn đến nút gốc trong $T^{(t)}$. Vì thế, *i* không thể biết được bất kỳ bí mật nút nào trong $T^{(t+1)}$.

d) Trường hợp 4:

Với $(i \notin S^{(t)} \wedge i \notin S^{(t+1)})$, điều đó có nghĩa là *i* bị hủy trước thời điểm *t*. Từ giả thuyết, *i* không bao giờ biết bất kỳ bí mật nút nào trong $T^{(t)}$. Từ những thông điệp thu hồi khóa tại thời điểm $t+1$ được mã hóa bằng bí mật nút trong $T^{(t)}$, *i* không thể khôi phục lại những thông điệp thu hồi khóa. Vì vậy, *i* không bao giờ tính ra được bất kỳ khóa bí mật nút nào trong $T^{(t+1)}$.□

2. Bổ đề 2

Cặp người dùng thông đồng User_A và User_C bất kỳ không thể tính ra bất kỳ khóa bí mật chưa biết nào từ OFT-1.

Chứng minh

Đầu tiên, xét User_C gia nhập hệ thống sau khi User_A rời khỏi nhóm. Trở lại Hình 3, giả sử User_A bị hủy tại thời điểm t_A sau đó User_C gia nhập hệ thống tại thời điểm t_C. Theo OFT-1 thì toàn bộ các bí mật nút mù của anh em theo đường dẫn của User_C đến nút gốc bị thay đổi tại thời điểm t_C. Vì thế, User_C không thể biết bất kỳ bí mật nút mù nào đã sử dụng trước khi nó gia nhập hệ thống. Vì vậy, User_A và User_C không thể tìm ra bất kỳ khóa bí mật chưa biết nào.

Tiếp theo, xét User_A và User_C gia nhập hệ thống cùng thời điểm. Theo OFT-1, toàn bộ các khóa bí mật nút trên đường dẫn của User_A và User_C bị thay đổi. User_A và User_C không thể tìm được bất kỳ khóa bí mật nút chưa biết nào bằng những bí mật nút mù mà chúng đã có. Trong trường hợp User_A và User_C cùng bị hủy tại một thời điểm cũng tương tự. Tổng hợp lại, User_A và User_C không thể tìm ra bất kỳ khóa bí mật nút chưa biết nào từ OFT-1.

3. Bổ đề 3

Giao thức OFT-1 là an toàn chống lại tấn công thông đồng.

Chứng minh

Giả sử tồn tại $t_0 \geq 0$ và một tập các kẻ tấn công không có trong $S^{(t_0)}$ có thể nhận được một bí mật nút x_i trong $T^{(t_0)}$. Vì OFT-1 là an toàn chống tấn công người dùng đơn nên mỗi kẻ tấn công không bao giờ nhận được x_i trong $T^{(t_0)}$. Những kẻ tấn công này phải thông đồng với nhau để nhận được x_i. Theo định lý 1 của Liu [20], phải tồn tại một cặp tấn công có thể thông đồng nhau để tính ra khóa bí mật nút chưa biết. Điều này mâu thuẫn với bổ đề 2. Vì vậy, giao thức OFT-1 là an toàn chống tấn công thông đồng.

VI. KẾT LUẬN

Mạng ĐHGSCN là một hệ thống rất quan trọng trong các cơ sở hạ tầng của quốc gia. Tuy nhiên, do sự phát triển của công nghệ mà mạng này phải đối mặt với những hiểm họa mất an toàn. Hậu quả từ mạng ĐGGSCN mất an toàn để lại rất lớn, nó có thể ảnh hưởng tới cuộc sống của người dân, tồn vong của một quốc gia. Bài báo tập trung vào xây dựng lược đồ quản lý và phân phối khóa an toàn và hiệu quả để đáp ứng yêu cầu sử dụng mật mã trong bảo vệ quá trình truyền dữ liệu trong mạng. Với lược đồ OFT-1 đề xuất, nó có thể chống lại tấn công phát lại nhờ khóa phiên sử dụng hàm băm mật mã với các tham số đầu vào là khóa, đánh dấu thời gian kết hợp với chuỗi số. Ngoài ra, OFT-1 chống tấn công thông đồng qua việc cập nhật lại toàn bộ những thông tin bí mật mà một SUB-MTU, RTU, MTU lưu trữ khi một thành viên mới gia nhập hoặc hủy một thành viên ra khỏi hệ thống. Qua đó cũng đảm bảo được tính bảo toàn bí mật trước và bảo toàn bí mật sau.

VII. TÀI LIỆU THAM KHẢO

[1] IEC. Technical Specification, Power systems management and associated information exchange - Data and Communications Security-Part 5: Security for IEC 60870-5 and derivatives, IEC Standard 62351, 2009.

[2] C. Beaver, D. Gallup, W. Neumann, & M. Torgerson, "Key management for SCADA," Technical report, Sandia, 2002.

[3] Robert Dawson, Colin Boyd, Ed Dawson, Juan Manuel Gonzalez Nieto, "SKMA A Key Management Architecture for SCADA Systems," In Proc. Fourth Australasian Information Security Workshop, Vol. 54, pp. 138-192, 2006.

[4] A.T. Sherman, D. A. McGrew, Key establishment in large dynamic groups using one-way function trees, IEEE Trans. Softw. Eng 29 (5), pp. 444-458, 2003.

[5] D. Balenson, D. McGrew, A. Sherman, Key Management For Large Dynamic Groups: One-Way Function Trees and Amortized Initialization, Internet Research Task Force, 2000.

[6] D. M. Wallner, E. J. Harder, R. C. Agee, Key Management for Multicast: Issues and Architectures, Internet Engineering Task Force, 1998.

[7] C. K. Wong, M. Gouda, S. S. Lam, Secure group communication using key graphs, IEEE/ACM Trans. Netw. 8 (1) (2000) 16-30 .

[8] G. Horng, Cryptanalysis of a key management scheme for secure multicast communications, IEICE Trans. Commun E85-B (5), pp.1050-1051, 2002.

[9] M. S. Hossain, Cloud-supported cyber-physical localization framework for patients monitoring, IEEE Syst. J, 2016.

[10] M. S. Hossain, G. Muhammad, Cloud-assisted industrial internet of things (IIot)- enabled framework for health monitoring, Comput. Netw, doi: 10.1016/ j.comnet.2016.01.009, 2016.

[11] C. H. Liu, B. Zhang, X. Su, J. Ma, W. Wang, K. K. Leung, Energy-aware participant selection for smartphone enabled mobile crowd sensing, Syst. J. PP (99) 1-12, 2015.

[12] C. H. Liu, J. Fan, H. Pan, J. Wu, K. K. Leung, Toward qoi and energy efficiency in participatory crowdsourcing, IEEE Trans. Veh. Technol. 64 (10), pp. 4684-4700, 2015.

[13] C. H. Liu, J. Fan, J. Branch, K. K. Leung, Toward qoi and energy-efficiency in internet-of-things sensory environments, IEEE Trans. Emerg. Topics Comput. 2 (4), pp. 473-487, 2014.

[14] C. H. Liu, B. Yang, T. Liu, Efficient naming, addressing and profile services in internet-of-things sensory environments, Ad Hoc Netw. 18, pp. 85-101, 2014.

[15] C. H. Liu, J. Zhao, H. Zhang, S. Guo, K. K. Leung, J. Crowcroft, Energy-efficient event detection by participatory sensing under budget constraints, Syst. J. PP (99) 1-12, 2016.

[16] B. Zhang, Z. Song, C. H. Liu, J. Ma, W. Wang, An event-driven qoi-aware participatory sensing framework with energy and budget constraints, ACM Trans. Intell. Syst. Technol. 6 (3) 42, 2015.

[17] K. He, J. Chen, R. Du, Q. Wu, G. Xue, X. Zhang, Deypos: deduplicatable dynamic proof of storage for multiuser environments, IEEE Trans. Comput., 2016.

[18] W. C. Ku, S. M. Chen, An improved key management scheme for large dynamic groups using one-way function trees, in: Proceedings International Conference Parallel Processing Workshops, Kaohsiung, Taiwan, pp. 391-396, 2003.

[19] X. Xu, L. Wang, A. Youssef, B. Zhu, Preventing collusion attacks on the one-way function tree (OFT) scheme, in: Proceedings 5th International Conference Applied Cryptography and Network Security, Zhuhai, China, pp. 177-193, 2007.

[20] J. Liu, B. Yang, Collusion-resistant multicast key distribution based on homomorphic one-way function trees, IEEE Trans. Inf. Forensics Security 6 (3) pp. 980-991, 2011.

[21] S. Panjwani, Private group communication: two persepectives and a unifying solution, Computer Science Engineering Department, University of California. San Diego, CA, 2007.

IMPROVEMENT METHODOLOGY OF SECURE KEY DISTRIBUTION IN RESTRICTED RESOURCE NETWORKS

Nguyen Dao Truong, Le My Tu

ABSTRACT: *This paper presents an approach to improving the security key distribution scheme in industrial supervisory and monitor networks with the aim of combating typical attacks such as replay attacks, collusion attacks when a member joins and two or more members join the system. Nowadays, This network due to requirements that have to contact a lot with public networks such as the Internet, WAN, the risk of unsecurity increasingly exist. These attacks can greatly affect the system, which can destroy the entire system and leave serious consequences not only to the organization itself but also to security of the country. Improvements in articles ensure security with minimal increase in computing costs.*