東海大學資訊工程研究所

碩士論文

指導教授：呂芳懌 博士

一個高效安全且具備緊急例外處理之救護車交通控制系統

An efficient and secure traffic control system with emergency handling for ambulances

研究生：陳詩涵

中華民國 一零四 年 六 月

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 陳 詩 涵 所提之論文

一個高效安全且具備緊急例外處理之救護車交通控制系統

經本委員會審查，符合碩士學位論文標準。

學位考試委員會
召 集 人 羅濟群 簽章

委 員 陳金鈴

指 導 教 授 簽章

中華民國 104 年 6 月 29 日

中文摘要

在醫療系統中，運送病人到醫院以挽救他們的生命是一項重要的任務，研究顯示，救護車的響應時間和患者的死亡率有密切的關係。響應時間較長，死亡率較高。在許多國家，實施緊急醫療服務之標準表明，在城市地區， 90％的救護車請求，救護車須在7~10分鐘到達現場。在本文中，我們提出了一種新的交通管制方案，稱為高效安全的道路控制系統（簡稱ESTCS），它可以引導救護車安全行車，並縮短其救援時間，和快速運送病患至醫院。本道路控制系統提供異常處理機制，當救護車遇到意外情況時，它可以快速，安全地解決這些問題。此外，ESTCS使用動態時間加密金鑰和二維串流加密函數，為用戶提供更安全的環境和更好的傳輸性能。

關鍵字:救護車，區域交通管理局，二進制加法，交通號誌控制，資料鏈結核心

Abstract

In a healthcare system, transporting patients to hospital is an important task for saving their lives since research indicates that there is a close relationship between ambulance response time and patient mortality. The longer the response time is, the higher the mortality will be. In many country, implemented standard of the emergency medical service shows that in urban areas, 90% of requests must be satisfied within 7~10 minutes. But some unexpected traffic situations may delay the rescue task, therefore, in this paper, we propose a novel traffic control scheme, called the Efficient and Secure Traffic Control System (ESTCS for short), aiming to guide an ambulance (AMU) to safely navigate so as to shorten its attendant time, and fast deliver patients to hospitals. The ESTCS also employs an exception handling mechanism, with which when an AMU encounters unforeseen situations, it can accordingly resolve them quickly and safely. In addition, a dynamic timing encryption key and a two-dimensional stream encryption function are employed by the ESTCS to provide AMUs with a more secure working environment and improve better performance.

Keywords: *Ambulance, RTA, Binary adder, Traffic light control, DCC*

List of Contents

List of Figures

List of Tables

1. Introduction

In a healthcare system, transporting patients to a hospital is an important task for saving their lives since research [1-4] indicates that there is a close relationship between ambulance (AMU for short) response time and patient mortality. The longer the response time is, the higher the mortality will be. Generally, the response time of an emergency medical services (EMS for short) is defined as the interval from the time when the corresponding call was received by the EMS provider to the time when the AMU arrives at the emergency scene [5-8]. In Spain, a reduction of the time interval between the crash happening and the arrival of the emergency services to the crash scene strongly lower the probability of death [9]. Lim et al. [10] claimed that a 10 min reduction of the medical response time has statistically associated with an average decrease of the probability of death by one third, both on motorways and conventional roads. In Montreal, Canada, the implemented standard for AMUs used by "Urgences Sante" states that 90% of requests should be served within 7 minutes [11]. The implemented standard of the United States Emergency Medical Service Act shows that in urban (rural) areas, 95% of AMU requests must be satisfied within 10 (30) minutes [12]. In U.K., 75% calls must be served within 8 minutes [13].

In a dense area, during rush hours, roads are often congested, very usually sticking AMUs on its way to destination in traffic jam, and consequently delaying the delivery of patients to the hospital. Studies [14-15] also indicate that compared with general vehicles, emergency vehicles, such as AMUs, fire trunks, police cars or other emergency response organizations' vehicles, are threatened with 8 times higher risk of being evolved in traffic accidents with severe accidents, and 4 times higher risk of being involved in fatal accidents. [15] also shows that each typical emergency trip in average takes 7 minutes long, and passes 4 red and 3 green traffic lights. As observed at specific road intersections, every red light will induce 15 to 30 seconds of extra delay. Wrong actions of other road users often cause another delays. In average, 2.5 drivers do not respond properly, resulting in about 1 minute of total delay per trip

[15].

Therefore, in this study, we propose an efficient traffic control scheme, named the Efficient and Secure Traffic Control System (ESTCS for short), which aims to guide an AMU to navigate on its way so as to safely and rapidly deliver patients to hospitals, and in which information is exchanged between regional transportation authority (RTA for short) and AMUs so that the RTA can effectively monitor the statuses of AMUs and control traffic lights for them. Compared with the system proposed in [16], the ESTCS provides an AMU with an exception handling function with which the AMU can accordingly solve the unforeseen situations that it meets on its way to the destination. Also, we employ a dynamic timing encryption key and a two-dimensional operator, to encrypt private keys to more effectively secure the ESTCS communication, and more efficiently accelerate its information exchange process.

2. Background and Related Work

2.1 VANET

Vehicular Ad-hoc Networks (VANETs) are a form of mobile ad-hoc network which provides users with vehicle-to-vehicle and vehicle-to-infrastructure communications and convenient wireless network services. VANET treats all participating vehicles as wireless routers or nodes, allowing them to connect to each other in a distance of approximately 100 to 300 meters and then creating a network of a wide range [17]. Vehicles installed with VANET can exchange traffic information and road condition (e.g., an accident on a road) with each other [18-19], implying that through the system, users can be notified with the road information immediately, and receive many extra network services and applications to increase their convenience and driving safety [20].

But a VANET architecture with respect to AMUs is not strictly protected when AMUs send messages to their front and/or near vehicles. Hence, security level of the system needs to be enhanced to prevent it from being cracked or compromised. Also, an AMU needs a mechanism to identify sender's identity so as to confirm correctness and integrity of a receiving message, meaning the encryption mechanisms of current VANET architecture need to be improved.

2.2 RSU

A Roadside Unit (RSU) is one of the essential infrastructural components of a VANET used to collect traffic data [21], particularly for collecting that of road intersections since road intersections are important parts of roads that may affect nearby road traffic. In addition, RSUs are connected to neighbor RSUs for easily exchanging traffic information or messages with each other, aiming to provide vehicle drivers with a convenient method for accessing network and traffic services.

Choi et.al [22] shows that RSUs as one of the security mechanisms can be used to authenticate and check integrity of receiving messages. However, this causes at least two problems. 1) Vehicles are unable to authenticate with each other when they stay at different RSUs' communication ranges since different RSU use different secret keys; 2) When a vehicle moves from one RSU range to another, the destination RSU requires the vehicle's certificate for source authentication and then negotiates a new secret key. This authentication may expose the certificate to hackers in the negotiation stage.

Therefore, in this study, we do not use RSU to deliver messages. Instead, a 3G/4G wireless system is employed, with which RTA and AMU exchange messages directly to make the system effectively achieve the stage of high safety and efficiency.

2.3 Binary Adder

The binary adder, denoted by +2, is a new encryption operator [23]. It adds two binary numbers of the same length. When encrypting data *D*, it binary-adds *D* and the encryption key *Q*, and ignores the overflow bit. The result is denoted by *D*'. To decrypt the data, if *Q* is equal or smaller than D', the adder directly subtracts *D*' by *Q*. Otherwise, it adds the two’s complement of *Q* to *D*'. Also, assume that both *D* and *Q* are m bits in length. The probability p of recovering the values of (*D*, *Q*) from *D* +2 *Q* on one trial is p = $\frac{1}{2^{\mathrm{m}}}$ [23]. This encryption operator provides a new choice of encrypting data. In fact, if we employ both the binary adder and exclusive-or operations to encrypt/decrypt data, the security level of the underlying system will be higher.

2.4 Data Connection Core (DCC)

Based on communication security, wireless systems have two basic characteristics. The first is that wireless messages delivered are insecure since both hackers and legitimate users can receive them. The other is that a communication management system has to check the identities of those active communicators to see whether they are legitimate users or not. If no links between the communication management system and a user are pre-established, then (1) at the beginning of the communication, they cannot safely communicate with each other to establish a secure channel; (2) they are unable to confirm the legality of each other's identity. This will cause many serious problems, such as being hacked to charge extra communication cost or leaking private data to others [24].

For this, some security parameters for securing the communication between the system and each individual user, called Data Connection Core (DCC for short), are built before their wireless communication starts, aiming to protect the initial negotiation of their key exchange process [23]. The details of DCC of this study will be defined later.

2.5 The related studies

RSA [25], as an asymmetric encryption system that uses public key and private key to encrypt and decrypt data, is one of the first practical public-key cryptosystems and has been widely used for securing data transmission. Public key used for encrypting data is a key known to the public, while private keys utilized by decryption side for data decryption is a secret key. The security of RSA algorithms relies on "difficulty of prime number’s decomposition." The bigger the number, the longer it takes time to decompose. Basically, the RSA algorithm can be solved if the cracking time is long enough. But hackers need to consider whether the cost is worth or not. In [26] by Chen et.al, the RSA algorithm is used to generate a key pair, e.g., by network entity x. The key pair is utilized as a session key to sign message *M* which is delivered between x and another network entity y. Also the session key used for communication between RTA and hospital includes RTA’s private key, the session key employed for communication between RTA and ambulance includes a pseudo identity and these keys are not known to others. Therefore, the scheme can ensure session-key security.

Leu et al. [16] proposed a DCC of the format (*AMUID*, k_i, e_i, d_i, N_i, *Cellphone No*) to encrypt initial messages sent for establishing a secure channel between two entities of a network connection, where k_i is an encrypting key, (e_i, d_i, N_i) is the RSA-triple keys in which e_i is the RSA encryption key, d_i is the RSA decryption key, and N_i is the RSA individual positive integer, and *Cellphone No* is the AMU’s cell phone number through which AMU can communicate with RTA.

PKI [27] allows users to encrypt data with a public key, and one of the users can decrypt the data by using the corresponding private key. Generally, PKI needs client software, server-side software, hardware on both sides, legal contracts, guaranteed operating programs and other components before it can be built. The signer's public key may be given by a third party for validating the signer's digital signature.

Basically, PKI assists its participants to achieve confidentiality, message integrity and user authentication without having to exchange any secret information in advance. However, PKI has some problems in practical use, such as uncertainty of certificate revocation and conditions of certificate issuance on credentials center, laws change, etc. They increase the difficulty and complexity of issuing credentials [28]. [29-30] utilized PKI and digital signatures to secure their vehicular networks. But they do not provide any mechanisms for certificate revocation.

Trusted platform module (TPM) [31] as a hardware chip with encryption capabilities is embedded in a vehicle to solve the problems of communication security and anonymity in VANET. TPM consists of many cryptographic engines: Asymmetric (ECC) [32], Symmetric, Random Number Generator (RNG), and Hash to encrypt/decrypt messages.

Wagan et al. [33] proposed a TPM-based security architecture which is used by trusted groups with the assistance of a PKI security mechanism. When a node is selected as a group leader, TPM randomly selects a symmetric key from pre-loaded set of keys by using Symmetric engine and RNG. The generated symmetric key is first converted into hash value, then it is encrypted with Attestation Identity Key [34] of the group leader, and shared among the group members via onboard unit (OBU).

Sun et al. [35] introduced ID-based encryption for pseudonym generation and identity authentication through a threshold scheme to satisfy security and privacy requirements. ID-based encryption can employ an physical IP address or the text-value of a domain name as a key to transform the key to a synonym with which to encrypt messages [36]. Huang et al. [37] proposed an authentication framework using the method of ID-based Signature (IBS) and the ID-based

Online/Offline Signature (IBOOS) for VANETs in order to reduce the computational cost for better performance. But vehicles generate one-time key every time before sending a message. This relatively consumes longer time. Second, the authority is regional, which means that every time the vehicle reaches a new region, it must register with authority before it can receive the required local services. Even et al. [38] presented an IBOOS scheme which enhances the efficiency of pairing process by separating signing process into an offline phase and an online phase, in which the verification is comparatively more efficient than that of IBS. There are many IBS and IBOOS schemes available for the proposed frameworks, mainly based on ECC and RSA signatures. In fact, most authors used RSA-based signatures. However, the size of a signature is large, considerably increasing message sizes. ECC-based signatures are also useful for signing and verifying messages and have short signature sizes. Therefore, for VANETs, ECC based signatures are considered more efficient than RSA signatures.

In VANETs, the offline phase can be executed initially at RSUs or vehicles, while the online phase is performed in vehicles during V2V communication [39] [40].

3. The ESTCS System Architecture

3.1 The Parameters

The parameters used by the ESTCS are defined and summarized below.

(1) RTA: Regional Transportation Authority

(2) *AMUID*: the identity of an AMU

(3) k_1, k_2: AMU's private keys used to protect those messages delivered between RTA and AMU

(4) k_e: a AMU's private key utilized by an exception handling process for protecting delivered exceptional messages

(5) *Cellphone-RTA*: the RTA's cellphone number, with which AMU can communicate with the RTA.

(6) *Cellphone-AMU*: the AMU's cellphone number, with which RTA can communicate with AMU.

(7) DCC: Data Connection Core which consists of 5 parameters, including AMUID, k_1, k_2, k_e and *Cellphone-AMU*

(8) *status*: the internal state of the security system, which shows the state that the ESTCS should achieve at the next step

(9) *OP-code*: the operation code which indicates the status of an encryption/a decryption process and the function of a wireless message

(10) t_{nonce}: the timestamp of current time

(11) *MDH*: an encryption of month, day and hour in a day is formatted by $MDH = Month + Day * Hour$

(12) K_{CT}: A Dynamic Time Key [41]. In our system, before sending a message, the system fetches cpu time, with which to generate K_{CT} as a time key to encrypt the message. K_{CT} consists of nanosecond, second, minute, and *MDH*, i.e., K_{CT}=nanosecond||second||minute||*MDH*||*MDH*||minute||second||nanosecond, where nanosecond is

9 digits long, *MDH* is 3 digits in length, each of the remaining items is 2 digits in length and each digit is 4 bits long, i.e., $|K_{CT}|$ = 9+2+2+3+3+2+2+9=32 digits = 128bits

(13) R_0~R_3: the random numbers generated by the RTA for encrypting delivered messages

(14) R_4: the random number generated by the RTA for encrypting transmitted messages in the exceptional handling stage. When the exception handling process that uses R_4 as a security parameter is completed, R_4 is discarded.

(15) A_0~A_6: random numbers generated by the AMU for encrypting transmitted messages

(16) flag: the value of a flag can be 0 or 1, indicating whether an AMU accepts an assigned task or not

(17) *LA*: the AMU's current location expressed by using longitude and latitude

(18) Route: the route from the AMU's current location to the accident scene or the designated hospital

(19) *Speed*: AMU's speed when it runs on the road. Speed is used to indicate road traffic.

(20) *T*: a time period, which indicates how often the AMU sends a message to inform RTA of its current location, i.e., *LA*

3.2 The Functions

The functions employed by the ESTCS are defined as follows.

(1) Exclusive-or operator ⊕ :

Encryption: $c = p \oplus K$, where *p* is plaintext, *K* is encryption key and *c* is generated ciphertext.

Decryption: $p = c \oplus K$.

(2) Binary-adder $+_2$:

Encryption: $c = p +_2 K$, where *p* and *K* undergo binary addition, and the overflow bit of adding *p* and *K* is ignored;

Decryption: $= c -_2 K = \begin{cases} c - K, & if\ c \geq K \\ c + \bar{K} + 1, & if\ c < K \end{cases}$,

where $-_2$ denotes the binary subtraction, and $\bar{K}$ is the one's complement of *K*.

(3) $f_{2D}(x;a,b)$: An encryption function defined as $f_{2D}(x;a,b)=(x \oplus a)+_2 b$, where *x*, *a*, and *b* are random

parameters generated by both RTA and AMU, individually.

$Invf_{2D}(y;a,b)=(y-_{2}b)\oplus a$ is the inverse function of $f_{2D}()$ where $y= f_{2D}(x;a,b)$.

(4) $f_{2D}s(String;a,b)$: An encryption function defined as $f_{2D}s(String;a,b)=CS_1//CS_2//CS_3...//CS_n$, where $String=S_1//S_2//S_3//...//S_n$, $CS_j = [S_j\oplus(a+_2CS_{j-1})]+_2(b\oplus S_{j-1})$, with $1\leq j\leq n$, $S_0 = a$, $CS_0=b$; a and b are encryption keys.

(5) $Invf_{2D}s(CStr;a,b)$: An decryption function defined as $Invf_{2D}s(CStr;a,b)=S_1//S_2//S_3//...//S_n$, where $S_j=[CS_j-_2(b\oplus S_{j-1})]\oplus(a+_2CS_{j-1})$, $1\leq j\leq n$, with $S_0 = a$, $CS_0=b$; a and b are decryption keys and $CStr= f_{2D}s(String;a,b)$.

(6) $HMAC(k)$: A Hash-based message authentication code generated by performing a hash function with secret key k on the transmitted message to ensure the certification and integrity of this message.

3.3 Databases

The databases utilized in this study are as follows.

(1) DCC database: A database which keeps all AMUs' DCCs.

(2) random-number database : A database which maintains all reserved random numbers as random keys.

(3) dynamic-record database : A database individually owned by both RTA and AMU for recording the status that a step of the proposed security process will achieve.

(4) event database : A database which holds all events having been occurred and finished.

3.4 OP-codes and Status Comparison Table

In the ESTCS, the *OP-code* and *status* at the first field of a message points out the function of the message. With the *OP-code* and *status*, both sides of communication can authenticate whether the message received is really sent by the other side or not.

Table 1. OP-codes and Status Comparison Table.

OP-code / *status*	functions
1	RTA assigns a task (the destination address and the optimal path) to AMU
2	AMU replies the task (to go / reject)
3	AMU returns the coordinates of its current location on the way to the accident scene
4	AMU arrives at the accident scene
5	RTA sends an optimal path to the designated hospital and the hospital address to AMU
6	AMU starts for the hospital
7	AMU on the way to the hospital returns the coordinates of its current location
8	AMU completes the task
9	Exception Handling --- an accident occurs on road
10	Exception Handling --- two AMUs orthogornally compete a traffic light
11	Exception Handling --- AMU has a flat tire or is out of work
12	preserved

3.5 System flow

The system flow of the ESTCS is illustrated in Figure1. The steps are as follows.

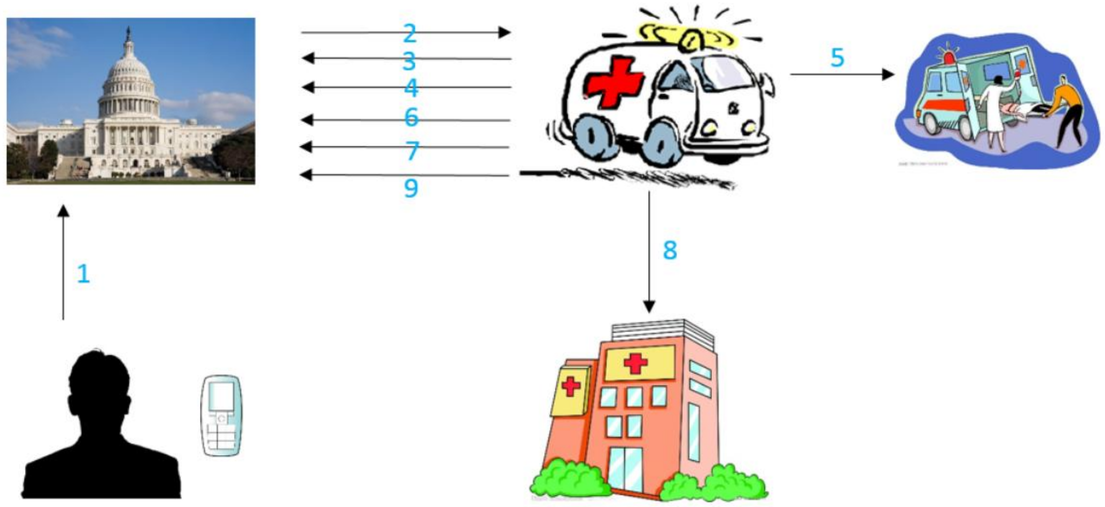

Figure 1. The system flow of the ESTCS.

Step 1: Informant → RTA: The informant who reports to RTA the accident information such as location, cause of accident, number of people injured, age, etc.

Step 2: RTA → AMU: RTA selects a suitable AMU and then asks the AMU to see whether it can go or not.

Step 3: AMU → RTA: AMU replies RTA whether it can go or not. If yes, AMU goes to the next step.

Step 4: AMU → RTA: AMU sends the coordinates of its current location to RTA periodically on the way to the accident scene.

Step 5: AMU → accident scene: When AMU arrives at accident scene, it notifies RTA to send hospital's information to it.

Step 6: AMU → RTA: AMU notifies RTA that it starts for the designed hospital.

Step 7: AMU → RTA: AMU sends the coordinates of its current location to RTA also periodically on the way to the hospital.

Step 8: AMU →hospital: AMU arrives at the hospital.

Step 9: AMU → RTA: AMU notifies RTA that it finishes this rescue mission.

3.6 Security between RTA and AMU

The security process between RTA and AMU is as follows.

Pre- step: RTA choosing an AMU

When RTA receives a phone call from someone, telling it there is an accident or a patient needs an AMU (in the following, an accident as an example), it requests the information of accident scene, like accident location, and status of a patient, judges the patient's injured severity and then chooses an appropriate AMU and requests the AMU to go to the accident scene.

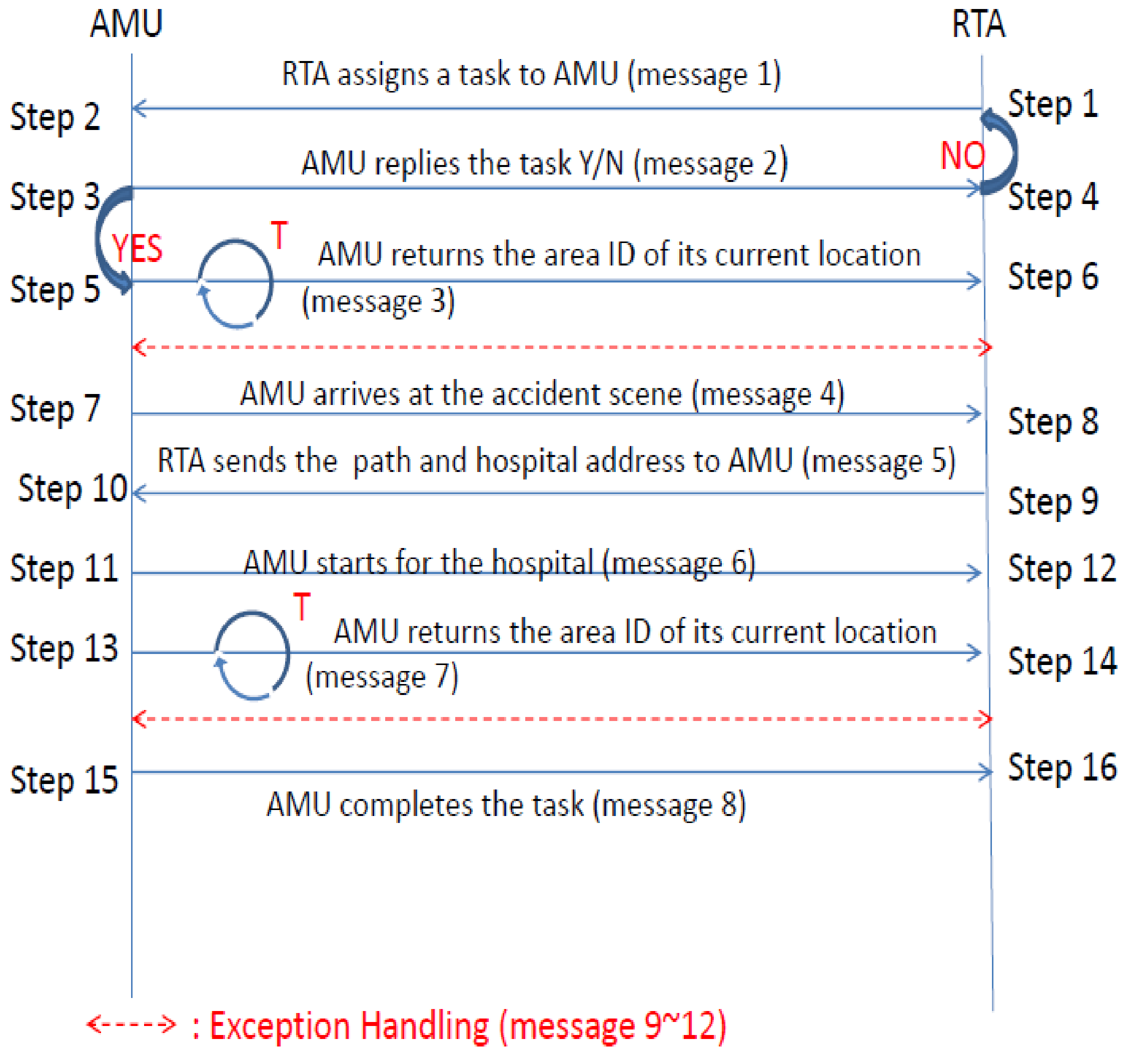
AMU
RTA
RTA assigns a task to AMU (message 1)
Step 1
Step 2
AMU replies the task Y/N (message 2)
NO
Step 3
Step 4
YES
T
AMU returns the area ID of its current location (message 3)
Step 5
Step 6
AMU arrives at the accident scene (message 4)
Step 7
Step 8
RTA sends the path and hospital address to AMU (message 5)
Step 9
Step 10
AMU starts for the hospital (message 6)
Step 11
Step 12
T
AMU returns the area ID of its current location (message 7)
Step 13
Step 14
Step 15
Step 16
AMU completes the task (message 8)
<----> : Exception Handling (message 9~12)

Figure 2. The security process between RTA and AMU.

T: AMU replies a message in Step 5/ Step 13 every T time period

Step 1: RTA -> AMU: RTA assigning the task to an AMU

After choosing an AMU, RTA first retrieves the DCC of the AMU from its DCC database, and stores the DCC as a dynamic record in the dynamic-record database. A dynamic record is used to keep track of the AMU's steps and status for this rescue task. RTA further

(1) randomly chooses four random numbers R_0~R_3 from its internal random-number database;

(2) fetches CPU time and derives t_{nonce}, which is then further invoked to generate K_{CT}, from the time;

(3) generates Message 1, the format of which is as follows;

$OP\text{-}code|t_{nonce}|[(R_0 \oplus k_1)+_2K_{CT}] \oplus [k_2+_2K_{CT}]|f_{2D}(R_1;k_1,R_0)|f_{2D}(R_2;k_2,R_1)|f_{2D}(R_3;R_1,R_2)|f_{2D}s(route//Cellphone\text{-}RTA//destination\ LA;R_1,R_2)|HMAC(R_0 \oplus R_3)$, in which *OP-code* = 1;

(4) sends Message 1 to AMU. Creates its dynamic record, the format of which is (*AMUID, status,* k_1, $k_{2,}$ k_e, R_0~R_3, *route, destination LA, Cellphone-RTA),* where k_1, $k_{2,}$ k_e and *Cellphone-AMU* are retrieved from DCC database. In this dynamic record, the *status* is set to 2.

Step 2: AMU: AMU decrypting the message

When receiving message 1, AMU

(1) verifies whether the *OP-code* is 1 or not. If not, it discards this message; Otherwise, it

(2) checks to see whether $t_{received} - t_{nonce} < \Delta T$ or not where ΔT is a predefined threshold. If not, it discards this message and waits for a valid one. Otherwise, it

(3) generates K_{CT} and decrypts $[(R_0 \oplus k_1)+_2K_{CT}] \oplus [k_2+_2K_{CT}]$ with K_{CT}, k_1 and k_2 to obtain R_0, i.e., $R_0=[T_k \oplus (k_2+_2K_{CT})]-_2K_{CT} \oplus k_1$ where $T_k=[(R_0 \oplus k_1)+_2K_{CT}] \oplus [k_2+_2 K_{CT}]$

(4) decrypts $(R_1 \oplus k_1)+_2R_0$ by using R_0 and k_1 to recover R_1 where $R_1=Invf_{2D}(R_1; k_1, R_0)$;

The processes of decrypting R_2 *and* R_3 are similar to that of decrypting R_1;

(5) verifies whether $HMAC(R_0 \oplus R_3)_c= HMAC(R_0 \oplus R_3)_r$ or not where subscript $c(r)$ means the expression is obtained by calculation (retrieved from the received message). If not, AMU discards

this message and waits for a valid one. Otherwise, it

(6) decrypts $f_{2D}s(route//Cellphone\text{-}RTA;R_1,R_2)$ by using R_1 and R_2 to recover route and *cellphone-RTA.*

Step 3: AMU->RTA : AMU replying RTA

In this step, AMU

(1) generates four random numbers A_0~A_3, and sends a message, denoted by Message 2, to RTA. The format of this message is as follows.

$OP\text{-}code|AMUID|flag|f_{2D}(A_0;R_0,R_1)|f_{2D}(A_1;R_1,R_2)|f_{2D}(A_2;R_2,R_3)|f_{2D}(A_3;R_3,A_0)|HMAC(A_1\oplus A_3)$

In this message, *OP-code*=2 and random numbers A_0~A_3 are protected by R_0~$R_{3.}$; *flag* may be 0 or 1. If *flag*=1, that means due to some reasons the AMU is inconvenient to accept the task. Otherwise, that is *flag*=0, indicating that AMU replies yes. The AMU then starts for the accident scene immediately, and

(2) creates its dynamic record

(AMUID, status, k1, k2, ke, R0~R3, A0~A3, route, destination LA , Cellphone-RTA) with a part of data carried in message 2, and status is set to 3.

Step 4: RTA: RTA decrypting Message 2

When receiving message 2, RTA

(1) checks to see whether the *OP-code* meets the RTA's current status (=2) or not. If not, RTA discards this message and waits for a valid one. Otherwise, it

(2) decrypts AMU's four encrypted random numbers A_0~A_3 by using R_0~R_3 where $A_0= Invf_{2D}(A_0; R_0, R_1)$. The processes of decrypting A_1~ A_3 are similar to that for decrypting A_0. RTA further

(3) verifies whether $HMAC(A_1\oplus A_3)_c= HMAC(A_1\oplus A_3)_r$ or not. If not, it discards this message and waits for a valid one. Otherwise, it

(4) checks *flag* field in Message 2. If it is 1, indicating that due to some reasons AMU cannot go, then the process goes back to Step 1 to look for another suitable AMU. Otherwise, meaning that *flag*=0 and AMU accepts the task, RTA

(5) updates its dynamic record

($AMUID$, $status$, k_1, $k_{2,}$ k_e, R_0~R_3, A_0~A_3, LA, $route$, $destination\ LA$, $Cellphone\text{-}AMU$) with a part of data carried in Message 2, If AMU replies yes, the *status* is set to 3; otherwise, the *status* is set to 1.

Step 5: AMU ->RTA: AMU returning LA of its current location

(1) In this step, AMU is now on the road and needs to return the LA of its current location to RTA periodically. The format of this message (denoted by Message 3) is shown below.

$OP\text{-}code|AMUID|i|speed|f_{2D}(LA;A_j,R_k)|HMAC(R_3+_2A_k\oplus R_j)$ in which i is a counter with the initial value of 1, $j = i \bmod 4$, *and* $k = (i+1) \bmod 3$.

(2) If AMU receives an emergency message from RTA, the format of this message is as follows.

$OP\text{-}code|t_{nounce}|(K_{CT}\oplus k_e)+_2(R_0\oplus R_4)|f_{2DS}(e\text{-}message;A_0,R_4)|HMAC((R_4+_2A_0)\oplus R_0)$;

If *OP-code*=9, then *e-message* in this message carries new_route. Otherwise, it means two or more ambulances want to pass intersection, lower priority of AMU have to wait higher priority of AMU pass, then traffic lights will turn to green. So the content of *e-message* is ”drive slowly”. AMU then

(A) verifies whether the *op-code* is 9 or 10. If not, it discards this message, and the process goes to Step 5-(3); Otherwise, it

(B) checks to see whether $t_{receive}$ -$t_{nounce}\leq\Delta T$. If not, it discards this message and the process goes to Step 5-(3). Otherwise, it derives K_{CT} from t_{nounce}; and

(C) decrypts $(K_{CT}\oplus k_e)+_2$ $(R_0\oplus R_4)$ to obtain R_4 and verifies whether $HMAC((R_4+_2A_0)\oplus R_0)_c$= $HMAC((R_4+_2A_0)\oplus R_0)_r$ or not. If not, it discards this message, and the process goes to Step 5-(3). Otherwise, AMU

(D) checks to see whether *OP-code*=9. If yes, it decrypts f_{2DS}(e-$message$; A_0, R_4) to obtain this *e-message* from which it further retrieves a new route and follows the new route to go to the destination. Otherwise, meaning *OP-code*=10, which represents that two AMUs compete a traffic light orthogornally.AMU slows its speed and waits for the traffic light to be green. Then process goes to Step 5-(1).

(3) If there is an exceptional event on AMU, such as flat tire or out of work on the road, AMU sends an emergency message to RTA. The format of this message is as follows.

$OP\text{-}code|AMUID|t_{nounce}|HMAC((A_2 \oplus R_2)+_2(K_{CT} \oplus k_e))$ in which $OP\text{-}code$=11. AMU initiates a timer ΔT_w to waits for RTA's confirming message. If AMU receives a confirming message from RTA before ΔT_w times out, and the message passes the corresponding verification, then it terminates this task. Otherwise, AMU resends the emergency message to RTA again;

(4) updates its dynamic record, i.e.,

($AMUID$, $status$, k_1, k_2, k_e, R_0~R_3, A_0~A_3, i, $speed$, LA, $route$, $destination\ LA$, $Cellphone\text{-}RTA$) with the new information carried in message 3 and $status$ is set to 3. AMU also

(5) checks to see whether current LA=$destination\ LA$ or not. If yes, the process goes to Step 7. Otherwise, i=i+1 and it goes to Step 5-(1) every time period T.

Step 6: RTA: RTA decrypting message 3

In this step, when receiving a message, RTA

(1) checks to see whether the $OP\text{-}code$ is 11 or not. If not, the process goes to Step 6-(2). Otherwise, it means that the message is an emergency one, the format of which is as follows.

$OP\text{-}code|AMUID|t_{nounce}|HMAC((A_2 \oplus R_2)+_2(K_{CT} \oplus k_e))$.

RTA then

(A) checks to see whether $t_{receive}$ -$t_{nounce} \leq \Delta T$. If not, RTA discards this message and waits for a valid one. Otherwise, it

(B) derives K_{CT} from t_{nounce}, and retrives DCC, based on $AMUID$, from its DCC database to obtain k_e;

(C) verifies whether $HMAC((A_2 \oplus R_2)+_2(K_{CT} \oplus k_e))_c$= $HMAC((A_2 \oplus R_2)+_2(K_{CT} \oplus k_e))_r$ or not. If not, it discards this message and the process goes to Step 6. Otherwise, RTA sends an emergency confirming message to AMU. The format of this message is as follows.

$\mathit{OP\text{-}code}|\mathit{AMUID}|t_{nounce}|\mathit{HMAC}((A_3 \oplus R_3)+_2(K_{CT} \oplus k_e))$

and then RTA finds another suitable AMU to restart the task.

(2) checks to see whether the *OP-code* is 3 or not. If not, RTA discards this message and waits for a valid one. Otherwise, it means this is Message 3, the format of which is

$\mathit{OP\text{-}code}|\mathit{AMUID}|i|\mathit{speed}|f_{2D}(LA;A_j,R_k)|\mathit{HMAC}(R_3+_2A_k \oplus R_j)$; RTA

(3) retrieves R_3, A_k, and R_j from the record stored in RTA's dynamic-record database based on the *AMUID*.

(4) verifies whether $\mathit{HMAC}(R_3 \oplus A_k+_2R_j)_c = \mathit{HMAC}(R_3 \oplus A_k +_2R_j)_r$ or not.

If not, RTA discards this message and waits for a valid one. Otherwise, it

(5) retrieves i, A_j, and R_k from the dynamic record;

(6) decrypts $f_{2D}(LA;A_j,R_k)$ to obtain LA where $LA = \mathit{Invf}_{2D}(LA; A_j , R_k)$;

(7) updates its dynamic record, i.e.,

(*AMUID*, *status*, k_1, k_2, k_e, R_0~R_3,A_0~A_3, *i*, *speed*, *LA*, *route*, *destination LA* , *Cellphone-AMU)*

with the data carried in Message 3 and *status* is set to 3.

(8) If there is an exceptional handling event, RTA sends an emergency message to AMU. If not, the process goes to Step 6-(9). The format of the message is as follows.

$\mathit{OP\text{-}code}|t_{nounce}|(K_{CT} \oplus k_e)+_2(R_0 \oplus R_4)|\ f_{2DS}(\mathit{e\text{-}message};A_0,R_4)\ |\mathit{HMAC}((R_4+_2A_0) \oplus R_0)$;

in which if *OP-code*=9, that means *e-message=new_route*. Otherwise, *e-message*="drive slowly", and then the process goes to Step 6-(8).

(9) the process goes to Step 6.

Step 7: AMU ->RTA: AMU arriving at accident scene

On arriving at the accident scene, AMU

(1) sends Message 4 to RTA. The format of this message is shown below with *OP-cod*e = 4, and AMU generates three random numbers A_4~ A_6 which are protected by R_0~R_2.

$OP\text{-}code|AMUID|f_{2D}(A_4;R_0,A_0)|f_{2D}(A_5;R_1,A_1)|f_{2D}(A_6;R_2,A_2)|HMAC(A_4 \oplus A_6)$;

(2) updates its dynamic record, i.e.,

(AMUID, status, k_1, $k_{2,}$ k_e, R_0~R_3, A_0~A_6, *i, speed, LA, route, destination LA, Cellphone-RTA),* in which *status* is set to 4.

Step 8: RTA: RTA decrypting Message 4

On receiving Message 4, RTA

(1) first verifies whether the *OP-code* is 3 or 4. If *OP-code* is neither 3 nor 4, RTA discards this message and waits for a valid one. If *OP-code* is 3, meaning AMU is still on its way to the accident scene, the process goes back to Step 6; If *OP-code* is 4, showing that AMU has arrived at the accident scene, RTA then

(2) decrypts A_4~A_6 by using R_0~R_2 and A_0~A_2 where $A_j=Invf_{2D}(A_j;\ R_{j\text{-}4}, A_{j\text{-}4})$, $4 \le j \le 6$.

(3) verifies whether $HMAC(A_4 \oplus A_6)_c = HMAC(A_4 \oplus A_6)_r$ or not. If not, RTA discards this message and waits for a valid one. Otherwise, the process continues.

Step 9: RTA -> AMU: RTA sending the optimal path and hospital's name and address to AMU. RTA

(1) sends Message 5, which carries hospital's name, phone number, address and destination (hospital) LA, to AMU. The format of Message 5 is as follows.

OP-code|f_{2DS}(*route//destination LA //hospital's name// hospital's phone number// hospital's address*;A_6,R_3)| $HMAC(A_4 +_2 A_5)$ with *OP-code*=5.

(2) updates its dynamic record (*AMUID, status,* k_1, $k_{2,}$ k_e, R_0~R_3, A_0~A_6, *i, speed, LA, route, destination LA, hospital's name, phone number, address, Cellphone-AMU*) with the new information carried in message 5 and *status* is set to 6.

Step 10: AMU : AMU decrypting messages

On receiving Message 5,

(1) AMU verifies this message by checking to see whether the *OP-code* carried in it is equal to the *status* (=5) kept in AMU's dynamic record or not, and verifies *HMAC*($A_4 +_2 A_5$). If the message

cannot pass both verifications, AMU discards this message and calls RTA to resend Message 5. Otherwise, AMU decrypts the route and hospital's information and starts for the hospital.

(2) AMU updates its dynamic record (*AMUID*, *status*, k_1, $k_{2,}$ k_e, R_0~R_3, A_0~A_6, *i*, *speed*, *LA*, *route*, *destination LA*, *hospital's name*, *phone number*, *address*, *Cellphone-RTA*), in which *status* is set to 6.

Step 11: AMU ->RTA: AMU starting for the hospital

AMU

(1) sends Message 6 to notify RTA of its departure. The format of Message 6 is as follows.

OP-code| *AMUID* |$HMAC((A_0 \oplus R_0)+_2A_3)$;

(2) updates its dynamic record in which *status* is set to 7.

Step 12: RTA : RTA decrypting Message 6

On receiving this message, RTA

(1) checks to see whether *OP-code*(=6) meets the *status* it keeps or not. If not, RTA discards this message and waits for a valid one from AMU. Otherwise, it

(2) verifies whether $HMAC((A_0 \oplus R_0)+_2A_3)_c = HMAC((A_0 \oplus R_0)+_2A_3)_r$ or not.

If not, it discards this message and waits for a valid one. Otherwise, RTA

(3) updates the dynamic record (*AMUID*, *status*, k_1, k_2, k_e, R_0~R_3, A_0~A_6, *i*, *speed*, *LA*, *route*, *destination LA*, *hospital's name*, *phone number*, *address*, *Cellphone-AMU*) with the information carried in Message 6 where *status* is set to 7.

Step 13: AMU ->RTA: AMU returning its current LA

(1) In this step, AMU is still on its way to the hospital and needs to send its current LA to RTA. The format of this message (denoted by Message7) is shown below.

OP-code| *AMUID* | *i* |*speed*| $f_{2D}(LA;A_j,R_k)$ |$HMAC(R_k \oplus A_j)$ in which *i* is a counter with initial value=1, $j = i$ mod 7, and $k = (i+1)$mod 4.

(2) If AMU receives an emergency message from RTA, the format is shown below.

OP-code |t_{nounce}| ($K_{CT}\oplus k_e$)$+_2$ ($R_0\oplus R_4$)| f_{2DS}(*e-message*;A_0,R_4)|*HMAC*(($R_4+_2A_0$)$\oplus R_0$)

in which if *OP-code*=9, then *e-message*=*new_route*. Otherwise, *e-message*=”drive slowly”. So AMU

(A) verifies whether or not *OP-code* is neither 9 nor 10. If yes, AMU discards this message, and the process goes to Step13-(3). Otherwise, indicating that *OP-code*=9 or 10, it further

(B) checks to see whether $t_{receive}$ -$t_{nounce}\leq \Delta T$. If not, it discards this message and the process goes to Step13-(3). Otherwise, it

(C) derives K_{CT} from t_{nounce} , and descryps ($K_{CT}\oplus k_e$)$+_2$ ($R_0\oplus R_4$) to obtain R_4.

(D) verifies whether *HMAC*(($R_4+_2A_0$)$\oplus R_0$)$_c$= *HMAC*(($R_4+_2A_0$)$\oplus R_0$)$_r$ or not. If not, it discards this message, and the process goes to Step13-(3). Otherwise, it

(E) checks to see whether *OP-code*=9. If yes, meaning there is a new route, it then decrypts f_{2DS}(*e-message*;A_0,R_4) to obtain the new route which will guide it to go to the hospital. Otherwise, it indicates that *OP-code*=10. AMU slows down its speed and waits for the traffic light to turn to green. Then process goes to Step 13-(1).

(3) If there is an exceptional event, such as flat tire or out of work on the road, AMU sends an emergency message to RTA. The format of this message is as follows.

OP-code|*AMUID*|t_{nounce}|*HMAC*(($A_2\oplus R_2$)$+_2$($K_{CT}\oplus k_e$)) in which *OP-code*=11. AMU initiates a timer ΔT_w to wait for RTA’s confirming message. If AMU receives a confirming message from RTA before ΔT_w times out, and the message passes the corresponding verification, it then stops this assigned task. Otherwise AMU resends the emergency message to RTA again.

(4) updates its dynamic record, i.e., (*AMUID*, *status*, k_1, k_2, k_e, R_0~R_3, A_0~A_6, *i*, *speed*, *LA*, *route*, *destination LA*, *hospital's name*, *phone number*, *address*, *Cellphone-RTA)* with the new information carried in Message 7, and *status* is set to 7. It also

(5) check to see whether current *LA*=*destination LA* or not. If yes, the process goes to Step 15.

Otherwise, i=i+1 and it goes to Step 13-(1) every time period T.

Step 14: RTA: RTA decrypting message

In this step, when receiving a message, RTA

(1) checks to see whether the *OP-code* is 11 or not. If not, the process goes to Step 14-(2). Otherwise, it means this message is an emergency one, the format of which is shown below.

OP-code|*AMUID*|t_{nounce}|$HMAC((A_2 \oplus R_2)+_2(K_{CT} \oplus k_e))$; RTA further

(A) checks to see whether $t_{receive}$ -$t_{nounce} \leq \Delta T$. If not, it discards this message and waits for a valid one. Otherwise, it

(B) derives K_{CT} from t_{nounce}, and retrives DCC from its DCC database based on *AMUID* to obtain the corresponding k_e;

(C) verifies whether $HMAC((A_2 \oplus R_2)+_2(K_{CT} \oplus k_e))_c$= $HMAC((A_2 \oplus R_2)+_2(K_{CT} \oplus k_e))_r$ or not. If not, it discards this message and the process goes to Step14. Otherwise, RTA sends an emergency confirm message to AMU. The format of this message is as follows:

OP-code|*AMUID*|t_{nounce}|$HMAC((A_3 \oplus R_3)+_2(K_{CT} \oplus k_e))$

Then the process goes to Step1, and RTA finds another suitable AMU to go to the accident scene;

(2) checks to see whether the *OP-code* is 7 or not. If not, RTA discards this message and waits for a valid one. Otherwise, it implies that it is Message 7, the format of which is as follows.

OP-code| *AMUID* | i |*speed*| $f_{2D}(LA;A_j,R_k)$| $HMAC(R_k \oplus A_j)$.

RTA consequently

(3) retrives A_j and R_k from the record stored in RTA's DCC database based on *AMUID*;

(4) verifies whether $HMAC(R_k \oplus A_j)_c$= $HMAC(R_k \oplus A_j)_r$ or not.If not, RTA discards this message and waits for a valid one. Otherwise, it

(5) further retrieves i, A_j and R_k from the corresponding record stored in RTA's DCC database based on *AMUID*;

(6) decrypts $f_{2D}(LA;A_j,R_k)$ to obtain LA where LA =$Invf_{2D}(LA; A_j , R_k)$;

(7) updates its dynamic record, i.e.,

(AMUID, status, k_1, $k_{2,}$ k_e, R_0~R_3, A_0~A_6, *i, speed, LA, route, destination LA, hospital's name, phone number, address, Cellphone-AMU)* with the information carried in Message 7 and *status* is set to 7.

(8) when there is an exceptional handling event, RTA sends an emergency message to AMU. If not, goes to Step 14-(9). The format of this emergency message is as follows.

$OP\text{-}code|t_{nounce}|(K_{CT}\oplus k_e)+_2(R_0\oplus R_4)|f_{2DS}(e\text{-}message;A_0,R_4)|HMAC((R_4+_2A_0)\oplus R_0);$

in which if *OP-code*=9, *e-message=new_route*. Otherwise, *e-message*=”drive slowly”, and then the process goes to Step 14-(8)

(9) The process goes to Step14.

Step 15: AMU ->RTA : AMU completing the task

On arriving at the hospital, AMU

(1) sends Message 8 to notify RTA of the completion of this task. In this message, *OP-code* is 8.

$OP\text{-}code|\ AMUID|\ HMAC((A_0\oplus R_3)\ +_2(A_6\oplus R_0))$

AMU further

(2) updates its dynamic record,

(AMUID, status, k_1, $k_{2,}$ k_e, R_0~R_3, A_0~A_6, *i, speed, LA, route, destination LA, hospital's name, phone number, address, Cellphone-RTA)* with the new information, and *status* is set to 1. At last, AMU stores all the information of the dynamic-record in its own dynamic record database.

Step 16: RTA: RTA decrypting Message 8

On receiving Message 8, RTA

(1) verifies whether *OP-code* is 8 or not, if not, RTA discards this message and waits for a valid one. Otherwise, it indicates that AMU has arrived at the designated hospital, and RTA further

(2) verifies whether $HMAC((A_0\oplus R_3)\ +_2(A_6\oplus R_0))_c=\ HMAC((A_0\oplus R_3)\ +_2(A_6\oplus R_0))_r$ or not. If not, it discards this message and waits for a valid one. Otherwise, RTA updates it’s dynamic record,

(AMUID, status, k_1, k_2, k_e, R_0~R_3, A_0~A_6, *i, speed, LA, route, destination LA, hospital's name, phone number, address, Cellphone-AMU)* with their current values. At last, RTA stores all the information in its own event database to finish the rescue task.

4. Security Analysis

In Message 1, the transmission of random variables R_0 ~R_2 by the RTA to AMU is well-protected, wherein R_0 is protected by k_1, k_2, K_{CT} and the two-dimensionally operator (i.e., $\oplus$and $+_2$). R_1 and R_2 are cryptographically protected by using $f_{2D}()$ function, i.e., $f_{2D}(R_1;K_1,R_0)$ and $f_{2D}(R_2;K_2,R_1)$, so that they can be transmitted securely. How they are well protected will be shown in Theorem 1.

Theorem 1:

In the ESTCS, assume a key is n-bits long. If Message 1 is captured by hackers, then the probability with which hackers can obtain the correct value of the transmitted random key R_0 is $\frac{1}{2^n}$. Furthermore, the probability with which to obtain the correct values of the transmitted random key R_1 and R_2 is $\frac{1}{2^{2n}}$.

<pf>

Let

$$s_0 = [(R_0 \oplus k_1) +_2 K_{CT}] \oplus (k_2 +_2 K_{CT}) \qquad (1)$$

If Message 1 is captured by hackers, s_0 is then known by them. In order to solve R_0, Eq. (1) can be rewritten as

$$R_0 = [(s_0 \oplus (k_2 +_2 K_{CT})) -_2 K_{CT}] \oplus k_1 \qquad (2)$$

Although K_{CT} can be derived from t_{nonce}, Eq. (2) still has two encryption keys, i.e., k_1 and k_2, which are unknown to hackers. Only who has the value of (k_1, k_2) pair can correctly solve R_0. But the probability with which to obtain correct (k_1, k_2) pair by hackers is $(\frac{1}{2^n})^2$. Hence, the probability to obtain R_0 by solving Eq. (1) is $(\frac{1}{2^n})^2$. However, $(\frac{1}{2^n})^2$is less than $(\frac{1}{2^n})$, which is the probability to obtain R_0 by a blind guess.

Moreover, hackers may collect the communication messages, especially Message 1, of some particular *AMUID* in order to crack the correspoding (k_1, k_2) pair. However, for each communication, the t_{nonce}

independently varies so that the corresponding time key K_{CT} is unique for each rescue task. Thus, each K_{CT} derived from an occurrence of (k_1, k_2) pair to encrypt R_0 is also unique, meaning that cracking R_0 by using s_0 by means of statistical analysis is almost impossible. This shows that the probability with which to obtain correct R_0 by hackers is $\left(\frac{1}{2^n}\right)$.

Furthermore, let

$$s_1 = f_{2D}(R_1;k_1,R_0) = (R_1 \oplus k_1) +_2 R_0 \qquad (3)$$

and

$$s_2 = f_{2D}(R_2;k_2,R_1) = (R_2 \oplus k_2) +_2 R_1 \qquad (4)$$

In fact, Eq. (3) and Eq. (4) can be rewritten to Eq. (5) and Eq. (6), respectively, i.e.,

$$R_1 = (s_1 -_2 R_0) \oplus k_1 \qquad (5)$$

$$R_2 = (s_2 -_2 R_1) \oplus k_2 \qquad (6)$$

If s_1 is known by hackers, Eq. (5) shows that the next step is obtaining R_1. But the correct values of R_0 and k_1 are unknown to hackers. Hence, the probability with which we can obtain correct value of R_1 is $\left(\frac{1}{2^n}\right)$. Similarly, since R_1 and k_2 are unknown to hackers and there is no way to crack Eq. (6). Thus, the probability to obtain correct value of R_2 is $\left(\frac{1}{2^n}\right)$. They together show that the probability with which to obtain correct values of both R_1 and R_2 by hackers is $\left(\frac{1}{2^n}\right) \times \left(\frac{1}{2^n}\right) = \frac{1}{2^{2n}}$.

In the ESTCS, a two dimensional stream cipher function, i.e., $f_{2DS}()$, is employed to protect the messages transmitted between RTA and AMU. $f_{2DS}()$ has high performance in encrypting a block of plaintext, e.g., S_j, into a block of ciphertext by using two operators, i.e., $\oplus$ and $+_2$, since their operation speeds are fast. $f_{2DS}()$ also has a very high security level and will be discussed in Theorem 2.

Theorem 2:

Let the encryption keys a, b and a plaintext block S_j, $1 \leqq j \leqq m$, in the ESTCS be n bits long. If the plaintext $S = S_1S_2S_3\ldots S_m$ is encrypted by the two dimensional stream cipher function, i.e., $f_{2DS}(S;a,b)$, and the function has been known by hackers, then the probability with which hackers can correctly obtain the value of S_j, $1 \leqq j \leqq m$, is $\frac{1}{2^n}$.

<pf>

Let

$$CS = f_{2DS}(S;a,b) = CS_1//CS_2//CS_3//\ldots//CS_{\mathrm{n}} \qquad (7)$$

where

$$CS_j = [S_j \oplus (a +_2 CS_{j\text{-}1})] +_2 (b \oplus S_{j\text{-}1}),\ 1 \leqq j \leqq m, \qquad (8)$$

and $S_0 = a$, $CS_{j0}=b$.

To decrypt CS_j and obtain S_j, Eq. (2) can be rewritten as.

$$S_j = [CS_j -_2 (b \oplus S_{j\text{-}1})] \oplus (a +_2 CS_{j\text{-}1}),\ 1 \leqq j \leqq m \qquad (9)$$

Then, for $j = 1$, we have

$$S_1 = [CS_1 -_2 (b \oplus a)] \oplus (a +_2 b) \qquad (10)$$

It indicates that in order to correctly obtain S_1 from known CS_1 by solving ing Eq. (10), the values of $(b \oplus a)$ and $(a +_2 b)$ are necessary. But hackers do not know the values of a and b, meaning that S_1 is well protected. Hence, the probability to correctly obtain S_1 from known CS_1 is $\frac{1}{2^n}$ [23].

For $2 \leqq j \leqq m$, Eq. (3) shows that only the values of $(b \oplus S_{j\text{-}1})$ and $(a +_2 CS_{j\text{-}1})$ are known before CS_j can be decrypted to obtain correct value of S_j. However, hackers do not known them. Further, the encryption keys of each block j, i.e., $(b \oplus S_{j\text{-}1})$ and $(a +_2 CS_{j\text{-}1})$, $1 \leqq j \leqq m$, are both derived from a sequential pseudo random number stream since S_j is derived from a, b, $S_{j\text{-}1}$, and $CS_{j\text{-}1}$ which undergo the two dimensional operation where a and b are two unknown encryption keys, $S_{j\text{-}1}$ as an independent varying unknown key is *a* sequential varying encryption key.

Thus, for each j, $1 \leqq j \leqq m$, the probability with which hackers can correctly obtain the value of each S_j from $CS_{j\text{-}1}$ is $\frac{1}{2^n}$ [23].

In a wireless communication environment, communication between two sides of a connection is not safe before a "channel key" is built. Even the channel key has been built, the security level at their initial communication is often low. In this study, before wireless communication starts, RTA and AMU are linked to each other through the DCC, and integrated with the dynamic time key K_{CT} and the two dimensional operation to protect the random key group, i.e., R_0~R_3 and A_0~A_6. Apart from this, the two dimensional stream function (i.e., f_{2DS}()) can also protect transmitted stream data, and so on. The first message sent, i.e., Message 1, has a very high security level. We will prove this in Theorem 3.

Theorem 3:

In the ESTCS, the first communication message between RTA and AMU, i.e., Message 1, can effectively defend three common attacks, i.e., including forgery attack, replay attack and eavesdropping attack.

<pf>

First, anyone, including a hacker, who does not have DCC, will not own the encryption keys k_1 and k_2, , implying that the encryption code, $[(R_0\oplus k_1)+_2K_{CT}]\oplus[k_2+_2K_{CT}]$, generated by hackers cannot be correctly decoded by the AMU. That is, $R_{0,h}\neq R_{0,A}$ where subscript h (A) represents that the random key R_0 is generated by hackers (obtained by the AMU through decoding). Similarly, $R_{1,h}\neq R_{1,A}$, $R_{2,h}\neq R_{2,A}$ and $R_{3,h}\neq R_{3,A}$ since R_1, R_2 and R_3 are sequentially derived from k_1, k_2 and R_0 through the two dimensional operation. Hence, the forged authentication code, i.e., $HMAC(R_0\oplus R_3)$, carried in Message 1 cannot pass the authentication performed by the AMU. That is, Message 1 can effectively defend the forgery attack.

Second, with the T_{nonce} and $HMAC(R_0\oplus R_3)$ carried in Message 1, the ESTCS can effectively defend a replay attack [42].

At last, hackers may collect a particular AMU's communication messages, especially collecting Message 1, for a period of time. However, for different communication sessions, T_{nonce}, K_{CT}, and R_0 ~ R_3 are individually changed randomly so that the values of encrypted codes $[(R_0\oplus k_1)+_2K_{CT}]\oplus[k_2+_2K_{CT}]$, $f_{2D}(R_1;k_1,R_0)$, $f_{2D}(R_2;k_2,R_1)$, $f_{2D}(R_3;R_1,R_2)$, and $f_{2D}s$(*route// Cellphone-RTA // destination LA*;R_1,R_2) carried in Message 1 vary randomly on different sessions. Thus, the statistical analysis on the collection of above encrypted codes is useless in cracking and obtaining the correct values of k_1 and k_2, indicating that Message 1 can effectively defend the eavesdropping attack.

In the ESTCS, Message 2 is sent by AMU to RTA. In this message, AMU uses linked random key R_0 ~ R_3 and the two dimensional encryption to sequentially protect random key A_0 ~ A_3. Furthermore, using A_1 and A_3 as *HMAC*'s encryption key enhances the security level of the hash message authentication code, i.e., *HMAC* ($A_1 \oplus A_3$), since this code is dynamically random and has significantly improved the safety of Message 2.

When mutually authenticating each other with Message 1 and Message 2, RTA and AMU establish a group of eight random keys, i.e., R_0 ~ R_3 and A_0 ~ A_3, between them to effectively combine channel keys (i.e., k_1 and k_2), the two dimensional operation (i.e., $\oplus$ and $+_2$), the two dimensional stream function (i.e., $f_{2DS}()$) and hash function (i.e., *HMAC*()). Thus, the ESTCS has a very high degree of safety and effectiveness. We will analyze the performance of the ESTCS in next section.

5. Performance analysis

In order to analyze the performance of the ESTCS, we simulated its key generation, and data encryption/decryption, and measured the time consumed by each of the generation and encryption/decryption steps. Table 2 shows the hardware specifications of the AMU and RTA test-bed. Each test was performed one hundred thousand times.

Table 2. The hardware specifications of our AMU and RTA test-beds.

Component	AMU	RTA
CPU	Intel i5 2.67GHz	Intel i7 3.07GHz
RAM	2GB	12GB
Platform	Windows 7, 32-bit	Windows 7, 64-bit

Table 3. The costs for key generations on AMU and RTA (- : does not exist).

Item	Key generation time (μs)					
	AMU(bits)			RTA(bits)		
	512	768	1024	512	768	1024
R_0~R_4	-	-	-	1.35	1.77	2.20
A_0~A_6	8.77	12.39	16.22	-	-	-
K_{CT}	3.43	3.43	3.43	2.22	2.22	2.22
$\oplus$	0.51	0.74	0.92	0.20	0.24	0.27
$+_2$	1.56	2.68	3.35	0.21	0.26	0.31
$-_2$	2.59	4.24	5.31	0.44	0.54	0.66
$f_{2D}()$	2.2	3.28	4.42	0.33	0.42	0.50
$f_{2DS}()$	6.27	9.48	13.06	0.73	0.95	1.17
$Inv_{2D}()$	3.25	4.9	6.09	0.57	0.71	0.83
$Inv_{2DS}()$	9.44	14.99	18.23	1.29	1.64	2.10
HMAC()	123.21	159.2	200.45	63.46	74.65	82.07

Table 3 shows the costs for key generations on AMU and RTA. Due to the hardware specifications, the costs of RTA are lower than AMU's. Moreover, we found that the cost increase of RTA on 512, 768, 1024 bits is not very significant, whereas the increase of AMU is obvious also due to the hardware's execution speed and memory capacity.

The numbers of operations required by the steps of the ESTCS are listed in Table 4.

R: random number generation

T: K_{CT} generation

X: exclusive-or

B: binary addition

S: binary subtraction

H: *HMAC* generation

Table 4. The operations that constitute each step of the ESTCS encryption/decryption process and their consuming times on key size=512, 768 and 1024 bits.

Step	Number of operations	Time consumed (μs)			Sender
		Size (bits)			
		512	768	1024	
1	*4R*+9*X*+8*B*+2*T*+*H*	72.46	84.75	90.96	RTA
2	9*X*+8*S*+2*T*+*H*	168.58	206.64	258.07	AMU
3	4R+5*X*+4*B*+*H*	135.55	180.7	227.72	AMU
4	5*X*+4*S*+*H*	66.22	78.01	86.06	RTA
5	*2X*+2*B*+*H*	127.35	166.04	208.99	AMU
6	*2X*+2*S*+*H*	64.74	76.21	83.93	RTA
7	3R+4*X*+3*B*+*H*	133.69	174.61	221.13	AMU
8	4*X*+3*S*+*H*	65.58	77.23	85.13	RTA
Total time consumed by Steps 1~8		**834.17**	**1044.19**	**1261.99**	**3140.35**
9	5*X*+6*B*+*H*	65.72	77.41	85.28	RTA
10	5*X*+6*S*+*H*	141.3	188.34	236.91	AMU
11	*X*+*B*+*H*	125.28	162.62	204.72	AMU
12	*X*+*S*+*H*	64.1	75.43	83	RTA
13	*2X* +*B* +*H*	125.79	163.36	205.64	AMU
14	*2X* +*S* +*H*	64.3	75.67	83.27	RTA
15	*2X*+*B*+*H*	125.79	163.36	205.64	AMU
16	*2X*+*S*+*H*	64.3	75.67	83.27	RTA
Total time consumed by Steps 9~16		**776.58**	**981.86**	**1187.73**	**2946.17**

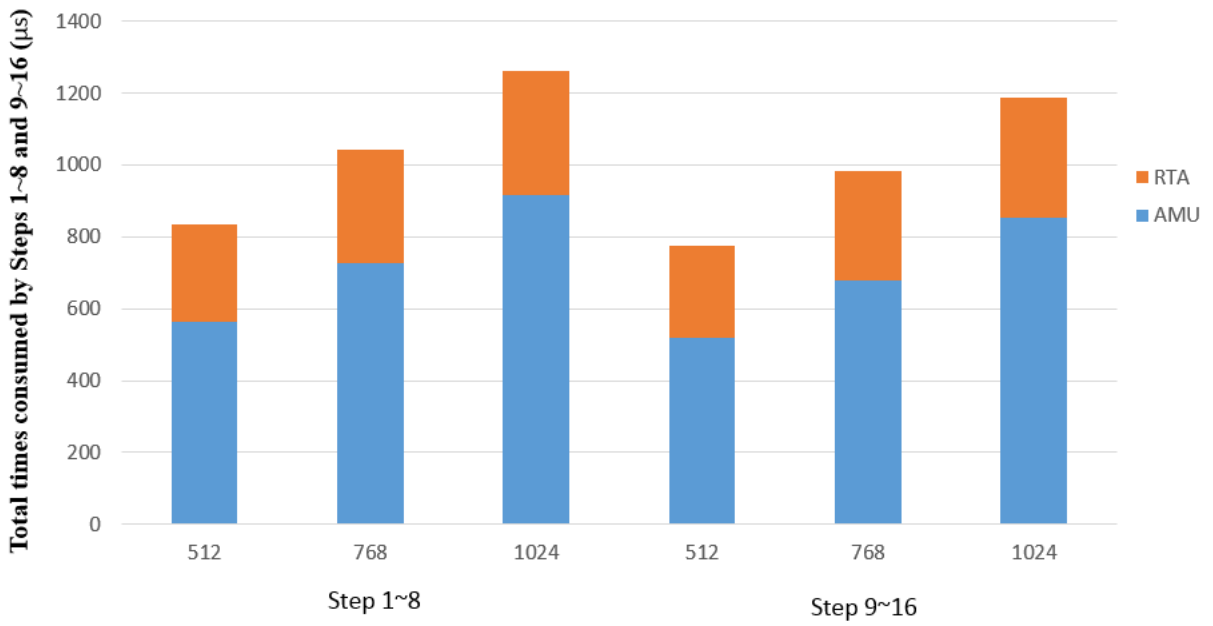

Figure 3. The accumulative times consumed by Steps1~8 and Steps 9~16 (μs)

Table 4 and Figure 3 show the times and accumulative times consumed by Steps 1~8 and Steps 9~16. The costs of the RTA are lower than those of the AMU, because RTA uses better hardware specifications than AMU does. Also, the time consumed by Step1 is the longest among those steps executed by RTA, including Steps1, 4, 6 and 8, since Step1 of RTA generates random keys, including R_0, R_1, R_2, R_3 and K_{CT}. Step2 on AMU has the same phenomenon because it decrypts them and store them as initial keys in its dynamic-record database. Now we divide the whole procedure of the ESTCS into two parts, Steps 1~8 as the first part which includes the activities performed by AMU and RTA when AMU is on its way to the accident scene. Steps 9~16 as the second part which are those activities that AMU and RTA have done when AMU is on its way to hospital from the accident scene. The time consumed by Steps 9~16 is a little shorter than that spent by Steps 1~8 because Step 1 takes a little longer time to generate random keys and K_{CT} highly protecting data accessed in the initial step since in a cryptographic system, this type of data is often more easily cracked by hackers than data accessed in the following steps. Further, Steps 9~16 only use pre-generated keys in the process, thus consuming a shorter time.

Table 5 lists the message delivery times through IEEE 802.11b wireless environment with the max data rate of 11Mbit/s [43]. Each message has 8-bit *OP-code* and 128-bit *HMAC*. t_{nonce} carried in Message 1 is 64 bits in length. The "Len" represents the chosen key length, and *F* indicates the length of a key which may be 512, 768, or 1024 bits long, for f_{2D}(). Basically, |Len|=|*F*|, Besides, *AMUID*, *flag* and *i* are 8 bits, and *speed* is 64 bits long.

Table 5. The lengths of messages generated by the ESTCS.

Transmission time consumed (μs)				
Message	Length (bits)	Media		
		802.11b (11Mbps)		
		512	768	1024
1	8+64+Len+6*F* +128	2.91	4.3	5.68
2	8+8+8+4*F*+128	1.69	2.48	3.27
3	8+8+8+64+*F* +128	0.56	0.76	0.95
4	8+8+3*F* +128	1.29	1.88	2.48
5	8+5*F* +128	2.08	3.06	4.05
6	8+8+128	0.11	0.11	0.11
7	8+8+8+64+*F* +128	0.56	0.76	0.95
8	8+8+128	0.11	0.11	0.11

We can see that Message 1 in Table 5 consumes the longest time since *F* contains Len and 6*F* in which Len and *F* are longer than 512 bits, 6*F* includes three f_{2D}()s and a f_{2DS}() where |f_{2DS}()|=*F* and |f_{2DS}()|=3*F* since *route//Cellphone-RTA//destination LA* will produce three concatenated encrypted data (see definition of f_{2DS}() in item 4 of section 3.2). The transmission time of Message 5 is also long because it contains hospital's information, including f_{2DS}(*route//destination LA //hospital's name// hospital's phone number// hospital's address*;A_6,R_3), which will generate five concatenated encrypted data. Messages 6 and 8 spend the least time because they are notification messages, with which AMU notifies RTA of its departure from the accident scene and the end of its rescue mission.

We also compared performance of the ESTCS with the systems developed by Leu [16] and Chen [26] for data encryption. Table 6 lists the numbers of operations required and times consumed by the three systems.

Table 6. The numbers of operations required and times consumed by the ESTCS, Leu's scheme [16] and Chen's approach [26] when encrypting/ decrypting data on key size = 512 bits. *x*/*y* stands for that a scheme requires to execute the corresponding operation for *x* times, totally consuming *y* μsec.

Operation	Time consumed (μs)	Scheme (times)		
		ESTCS	Leu's [16]	Chen's [26]
Exclusive-or	0.20	60/12	100/20	2/0.4
Binary addition	0.21	52/10.92	52/10.92	0/0
Symmetric encryption (AES-512)	55.241	0/0	0/0	56/3093.5
Asymmetric encryption	810	0/0	4/3240	7/5670
Hash function	63.46	16/1015.36	20/1269.2	12/761.52
Total time		1038.28	4540.12	9525.42

Table 6 shows that using an asymmetric encryption method to encrypt data often consumes a longer time than that required by a symmetric approach. In the ESTCS, as a symmetric encryption approach, we only use two operators exclusive-or and binary adder to encrypt data. So the ESTCS is more efficient than the other two tested systems.

6. Conclusions and Future Works

The main purposes of developing a secure traffic control system are giving an AMU a safe navigation environment, and shortening the attending time by considering exceptional handling events, thus safely and fast transporting patients to hospitals.

The ESTCS has the following features: (1) Using multiple random numbers, i.e., R_0~R_4 and A_0~A_6, to increase the safety of keys and data carried in a transmitted message. (2) Through the integration of AMU’s private keys (k_1, k_2) in DCC, the dynamic time key K_{CT} and a two-dimensional operator, the ESTCS converts (k_1, k_2) to a dynamic encryption key to effectively protect random keys (i.e., R_0~R_4, A_0~A_6) carried in transmitted messages. (3) Through the integration of AMU’s private key k_e, the dynamic time key K_{CT} and a two dimensional operator, the ESTCS safely and effectively protects emergency messages. (4) Using a two-dimensional stream function, i.e.,f_{2DS}(), the ESTCS quickly encrypts a plaintext stream into the corresponding ciphertext stream and safely protects the ciphertext stream carried in a delivered message.

And our contribution is (1) RTA builds AMU's DCC database which creates a closed-end security mechanism between RTA and AMU. (2) RTA and AMU deliver emergency messages through dedicated channels. So the delivery is safe and fast. (3) In order to handle exceptional situations, RTA keeps AMU’s individual dynamic record, which can effectively control the AMU’s status. (4) Exception handling of AMU is invoked to make the function of the ESTCS more complete, enforcing it to be more practically used. (5) *HMAC* function is employed to ensure the integrity and non-repudiation of transmitted messages. (6) *OP-codes* and *status* are integrated to enhance the security levels of transmitted messages, efficiency of data processing, and resistance of replay attacks.

To improve performance of the ESTCS, we abandon traditional security algorithms and functions, such as Diffie-Hellman [44], PKDS [45], RSA, and ECC, because they often require substantial computation, consequently causing low performance for a security system. Relatively, the ESTCS uses a two-dimensional stream encryption function and $f_{2D}()$ to securely protect delivered messages and data.

In fact, $f_{2D}()$ employs two different basic encryption operators, i.e., $\oplus$ and $+_2$, and two independent keys, i,e., k_1 and k_2, to highly raise its security level. Besides, $\oplus$ and $+_2$ operators are employed to encrypt data, and secret keys k_1 and k_2 are utilized to protect random number R_0~R_4, A_0~A_6 and dynamic time key K_{CT}, making the ESTCS more safe and accelerating its message processing speed.

Since the traffic condition is fast changing, if we can quickly grasp more detailed traffic information, it will be more helpful for the AMU to effectively, quickly and safely complete its rescue mission. So how to integrate VANET and RSU with the ESTCS will be an important task. We will also derive the formal reliability model and behavior model for the ESTCS so that users can predict its reliability and behaviors before using it. These constitute our future studies.

References

[1] R. Sánchez-Mangas, A. García-Ferrrer, A. de Juan, and AM Arroyo, "The probability of death in road traffic accidents. How important is a quick medical response?" *Accident Analysis & Prevention*, vol. 42, no. 4, pp. 1048-1056, July 2010.

[2] R.P. Gonzalez, GR Cummings, H.A. Phelan, M.S. Mulekar, and C.B. Rodning, "Does increased emergency medical services prehospital time affect patient mortality in rural motor vehicle crashes? A statewide analysis," *The American Journal of Surgery*, vol. 197, no. 1, pp. 30-34, Jan. 2009.

[3] R.B. Vukmir, "Survival from prehospital cardiac arrest is critically dependent upon response time," *Resuscitation*, vol. 69, no. 2, pp. 229-234, May 2006.

[4] "Part 12: From science to survival: Strengthening the chain of survival in every community," *Resuscitation*, vol. 46, no. 1-3, pp. 417-430, Aug. 2000.

[5] U.K. National Statistics, Ambulance services England 2008-2009, NHS Inform Center, 2009.

[6] M. Castrén, et al., "Recommended guidelines for reporting on emergency medical dispatch when conducting research in emergency medicine: The Utstein style," *Resuscitation*, vol. 79, no. 2, pp. 193-197, Nov. 2008.

[7] P.T. Pons and V.J. Markovchick, "Eight minutes or less: Does the ambulance response time guideline impact trauma patient outcome?" *Emergency Medicine Journal*, vol. 23, no. 1, pp. 43-48, July 2002.

[8] A.K. Marsden, "Getting the right ambulance to the right patient at the right time," *Accident and Emergency Nursing*, vol. 3, no. 4, pp. 177-183, Oct. 1995.

[9] J.F. Repede and J.J. Bernardo, "Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky," *European Journal of Operational Research*, vol. 75, no. 3, pp. 567-581, June 1994.

[10] C.S. Lim, R. Mamat and T. Bräunl, "Impact of ambulance dispatch policies on performance of

emergency medical services," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 624-632, June 2011.

[11] M. Gendreau, G. Laporte, and F. Semet, "A dynamic model and parallel tabu search heuristic for real-time ambulance relocation," *Parallel computing* vol.27, no.12, pp.1641-1653, 2001

[12] M.O. Ball and L.F. Lin, "A reliability model applied to emergency service vehicle location," *Operations Research*, vol. 41, no. 1, pp.18–36, Jan./Feb. 1993.

[13] J.J.M. Black and G.D. Davies, "International EMS systems: United kingdom," *Resuscitation*, vol. 64, no. 1, pp. 21–29, Jan. 2005.

[14] Bundesanstalt f¨ur Straßenwesen, "Ursachenuntersuchung inner¨ortlicher Unfallstellen," Wissenschaftliche Informationen der Bundesanstalt f¨ur Straßenwesen, 1994.

[15] A. Buchenscheit, F. Schaub, F. Kargl, and M. Weber, "A VANET-based emergency vehicle warning system", *IEEE Vehicular Networking Conference*, Tokyo, pp. 1-8, 2009.

[16] Y.L. Huang, I.L. Lin, F.Y. Leu, J.C Liu, F.C. Jiang, C.C. Chu, C.T. Yang, M.H. Chen, "A Secure Authentication System for Controlling Traffic Lights for Ambulances," Journal of Internet Technology, May 31, 2013.

[17] vehicular ad hoc network

http://en.wikipedia.org/wiki/Vehicular_ad_hoc_network

[18] U.S. Dept. Transp., "National highway traffic safety administration," *Vehicle Safety Communications Project*, 2006.

[19] S. Lee, G. Pan, J. Park, M. Gerla and S. Lu, "Secure incentives for commercial ad dissemination in vehicular networks," *ACM International Symposium on MobiHoc*, pp. 150-159, 2007.

[20] C. Zhang, X. Lin, R. Lu, P.-H. Ho and X. Shen, "An efficient message authentication scheme for vehicular communications," *IEEE Transactions On Vehicular Technology*, vol. 57, no. 6, pp. 3357-3368, Nov. 2008.

[21] J.Chi, et al., "An effective RSU allocation strategy for maximizing vehicular network

connectivity," *International Journal of Control & Automation* vol.6, no.4, pp.259-270, 2013.

[22] J.Y. Choi, M. Jakobsson, and S. Wetzel, "Balancing auditability and privacy in vehicular networks," *ACM International Workshop Q2S Winet*, pp. 79-87, 2005.

[23] Y.L. Huang, F.Y. Leu, and K.C. Wei, "A secure communication over wireless environments by using a data connection core," *Mathematical and Computer Modeling*, vol.58, no. 5-6, pp.

[24] M.K. Choi1, R.J Robles, C.H Hong, and T.H Kim, "Wireless network security: vulnerabilities, threats and countermeasures," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 3, no. 3, July 2008.

1459-1474, Sep. 2013.

[25] RSA algorithm

http://en.wikipedia.org/wiki/RSA

[26] C.L. Chen, I.C. Chang, C.H. Chang and Y.F. Wang, "A secure ambulance communication protocol for VANET," *Wireless personal communications*, vol.73, no.3, pp. 1187-1213. Dec. 2013.

[27] Public key infrastructure (PKI)

http://en.wikipedia.org/wiki/Public_key_infrastructure

[28] Common issues in PKI implementations http://www.sans.org/reading-room/whitepapers/authentication/common-issues-pki-implementations-climbing-slope-enlightenment-1198

[29] G.Samara, W.A.H. Al-Salihy, and R. Sures, "Security issues and challenges of vehicular ad hoc networks (VANET)," *New Trends in Information Science and Service Science*, pp. 393-398, May. 2010.

[30] G.Samara, W.A.H. Al-Salihy, and R. Sures, "Security analysis of vehicular ad hoc networks (VANET)." *Network Applications Protocols and Services*, pp. 55-60, Sep. 2010.

[31] Trusted Platform Module (TPM)

http://en.wikipedia.org/wiki/Trusted_Platform_Module

[32] Elliptic curve cryptography(ECC)

http://en.wikipedia.org/wiki/Elliptic_curve_cryptography

[33] A.A. Wagan, B.M. Mughal, H. Hasbullah, "VANET security framework for trusted grouping using TPM hardware," *Communication Software and Networks*, pp.309-312, 2010.

[34] Attestation Identity Key (AIK) Certificate Enrollment Specification Frequently Asked Questions

https://www.trustedcomputinggroup.org/files/resource_files/73942790-1A4B-B294-D0FE8D9D839D7A34/IWG%20AIK%20CMC%20enrollment%20FAQ.pdf

[35] J. Sun, C. Zhang, and Y. Fang, "An ID-based framework achieving privacy and non-repudiation in vehicular ad hoc networks," *Military Communications Conference*, pp.29-31, 2007.

[36] ID-based encryption

http://en.wikipedia.org/wiki/ID-based_encryption

[37] L. Huang, J. Li, and M. Guizani, " A novel ID-based authentication framework with adaptive privacy preservation for VANETs," Computing, Communications and Applications Conference, pp. 345–350, 2012.

[38] S. Even, O. Goldreich, and S. Micali, "On-line/off-line digital signatures," Cryptology-CRYPTO'89, LNCS, vol. 435, pp. 263–277, 1990.

[39] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): status, results, and challenges," *Telecommunication Systems* (Online First), pp. 1–25, 2010.

[40] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.

[41] Y.L. Huang, F.Y Leu, J.H Chen, William C.C Chu, and C.T Yang, "A true random-number encryption a method," Innovative Mobile and Internet Services in Ubiquitous Computing, pp.654-659, 2013.

[42] Replay attack

http://en.wikipedia.org/wiki/Replay_attack_

[43] IEEE 802.11

https://en.wikipedia.org/wiki/IEEE_802.11

[44] Diffie–Hellman key exchange

http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

[45] The Public Key Data Set (PKDS)

http://www-01.ibm.com/support/knowledgecenter/SSLTBW_1.13.0/com.ibm.zos.r13.csfb200/pkds1.htm%23pkds1