A Secure Authentication System for Controlling Traffic Lights for Ambulances

Yi-Li Huang[1], I-Long Lin[2], Fang-Yie Leu[1], Jung-Chun Liu[1], Fuu-Cheng Jiang[1], William Cheng-Chung Chu[1], Chao-Tung Yang[1]
[1]Department of Computer Science, TungHai University, Taiwan
[2]Department of Information Management, YuanPei University of Medical Technology, Taiwan
yifung@thu.edu.tw, cyberpaul747@mail.ypu.edu.tw, {leufy, jcliu, admor, cchu, ctyang}@thu.edu.tw

Abstract

When an ambulance (AMU for short) is going toward a hospital or an accident scene, if there is no traffic control and other guidance supports, due to a traffic jam or other reasons, the AMU may be unable to quickly arrive at the destination. Then, the time at which the hospital can start medically or surgically treating the injured people will be delayed. The earlier the people can be treated, the lower the mortality rate will be. To solve this problem, in this paper, we propose a traffic control scheme, called the AMU traffic control system (ATCS for short) , by which before an AMU passes through a street/road intersection, the ATCS turns the traffic lights to green so that the injured people can be transported to a nearby hospital as soon as possible. While the Regional Transportation Authority (RTA) and AMU communicate with each other, the transmitted messages are encrypted by random numbers and the RSA algorithm. According to our analyses, the system can effectively and efficiently protect the messages delivered through a wireless channel.

Keywords: Ambulance, Security, Traffic lights, RSA, OP-code table.

1 Introduction

In Taiwan, due to traffic accidents, more than 2,000 and 1,900 people died in 2010 and 2011, respectively. When a traffic accident occurs, one of the most important things to be dealt with is saving human's life. To achieve this, an ambulance (AMU for short) is often needed to transport injured people to the hospital. However, when the AMU is on its way, it may be stuck in a traffic jam. This will delay the people to be medically or surgically treated. If police can direct traffic or control traffic lights, the AMU can then be driven in smooth traffic so that the injured people can be sent to the hospital more rapidly. But this is infeasible, since polices cannot stand on the street to control traffic lights all day long.

On the other hand, there is a tight relationship between the AMU response time and the mortality rate [1-5]. The former is defined as the time period from the moment when an AMU request is received by the operator to the moment when the AMU arrives at the accident scene [6-10]. Some countries have implemented the standard of the response time, in which the AMU should arrive at the accident scene within a specific time period. For example, in Montreal, Canada, the implemented standard for AMUs run by "Urgences Santé" states that 90% of requests should be served within 7 minutes [11]. The implemented standard of the United States Emergency Medical Services Act [12] shows that in urban (rural) areas, 95% of AMU requests must be satisfied within 10 (30) minutes. In U.K., 75% calls must be served within 8 minutes, and 95% category-B (category-C) calls' response time should be less than 14 (19) minutes in urban (rural) areas [13]. However, in Taiwan, there are still no legal rules concerning the response time.

[14-20] proposed different methods to reduce the response time of an AMU. For example, in the AMU location models [14-17], AMUs stand by at specific locations so that they can arrive at the accident scene in a predefined time period. However, these models cannot solve the problem that the AMU may be stuck in a traffic jam.

[18-20] classified AMUs into two classes, one-tier and two-tier systems. In [18], a two-tier system providing basic life support and advanced life support [18] has better performance than that of an one-tier system. In [19-20], a two-tier system's cardiac arrest survival rate is higher than that of an one-tier system. Therefore, in this study, we propose a traffic control scheme, called the ambulance traffic control system (ATCS for short), which as a two-tier system turns the traffic lights of a street intersection into green right before the AMU arrives at the intersection so that the patients or injured people can be sent to a nearby hospital as soon as possible.

The scenario is that when an accident occurs, an informant calls the Regional Transportation Authority (RTA for short), which is an institute responsible for processing this type of requests, designating the most suitable AMU to serve the request, planning the route to the destination from AMU's current position, and controlling traffic lights along the route. In response, the RTA will retrieve the latitude and longitude of the accident scene based on the caller's description, and then requests the most suitable AMU to go. On receiving the request, the AMU leaves for the accident scene, and RTA starts controlling the traffic lights along

*Corresponding author: Fang-Yie Leu; E-mail: leufy@thu.edu.tw
DOI: 10.6138/JIT.2014.16.1.20130131

the route from the AMU's current position to the accident scene. RTA does the same when the AMU is going toward a nearby hospital from the accident scene.

The rest of this paper is organized as follows. Section 2 describes the background and related work of this study. Section 3 introduces a secure communication protocol that AMU employs to interact with RTA. Section 4 analyzes the security of the proposed system. Section 5 concludes this paper and outlines our future work.

2 Background and Related Work

Many studies have tried to reduce the response time of an AMU [14-17]. Most of them introduced specific AMU location models. Brotcorne et al. [14] presented two models, deterministic models and probabilistic models. The former is invoked during the planning stage of a rescue process for overlooking stochastic considerations regarding the usability of AMUs. The later simulates the behaviors of those AMUs unable to respond the calls by using a queuing system. Church and ReVelle [15] and Gendreau et al. [17] proposed the coverage maximization models that use a limited number of AMUs in the demand coverage. Toregas et al. [16] employed the minimum number of AMUs to cover all demands. In summary, these papers introduced different methods to describe AMU's location so as to reduce the response time of an AMU. But when the AMU was stuck in a traffic jam, these methods are not helpful.

Cheng [21] presented a model, in which AMUs, hospitals and a Road-side Transportation Authority (RSTA for short) were deployed. When a hospital, rather than RTA, receives an AMU-requesting call, it communicates with RSTA. RSTA then sends a session key ($SK_{A\text{-}RTA}$) to the hospital. The hospital passes the key to the AMU. After that, RSTA searches the shortest route to the accident scene from the AMU's current location, and sends the route to the AMU. When arriving at the accident scene, AMU sends the related information to RSTA. RSTA generates the shortest route to the hospital, and delivers the route to the AMU. On receiving this message, AMU starts for the hospital. But when AMU was stuck in a traffic jam, it could not rush to the accident scene or the hospital, either.

In this study, we propose the ACTS to dispatch the most suitable AMU to serve the rescue task, and control traffic lights to make traffic smooth so that the AMU can go to the accident and the chosen hospital through smooth traffic.

3 The AMU Traffic Control Scheme

The ATCS has six features. (1) The dispatched AMU is independent from the chosen hospital. The available AMU closest to the accident scene is enquired first. Then the most suitable hospital for providing the medical or surgical operations for the injured or patient is decided. The AMU and the hospital do not necessarily belong to the same medical organization. (2) The AMU reports its location to RTA periodically so RTA can precisely control the traffic lights to smooth traffic for the AMU. (3) An OP-code Table is established, through which the function of a wireless message can be identified and recognized. (4) A double authentication mechanism for wireless communication is securely and completely constructed. (5) Wireless messages delivered between RTA and AMU is more securely ensured and flexibly protected with the deployment of the double authentication mechanism. (6) For a rescue task, RTA provides the dispatched AMU with a private cell phone number so that the AMU and RTA can effectively handle unexpected situations through the cell phone.

3.1 System Flow Chart

The system flow chart of the ATCS is shown in Figure 1. Let's briefly describe it first.

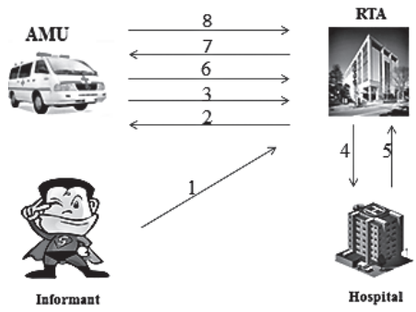

Figure 1 The System Flow Chart of the ATCS

Step 1: Informant → RTA: The informant notifies RTA of the information of the accident scene (longitude and latitude) and the condition of the injured. RTA then implements a rescue event according to the information.

Step 2: RTA → AMU: Based on the information of the injured condition and accident scene, RTA searches for a suitable AMU and the shortest path/route to the accident scene. After that, it sends a message, which contains the information of the injured, to the AMU, and enquires the AMU to see whether it is available to perform the rescue task or not.

Step 3: AMU → RTA: On receiving the task, the AMU responds with yes/no of its availability. If no, RTA repeats step 2. Otherwise the AMU returns a task response message to RTA, and starts the rescue task.

Step 4: RTA → hospital: Based on the information of the

injured described by the informant, RTA selects the closest hospital that meets the injured's need, sends the information of the injured condition to the hospital, and enquires its availability.

Step 5: Hospital → RTA: On receiving the message, the hospital responds with yes/no of its availability. If no, RTA searches for another suitable hospital by repeating step 4.

Step 6: AMU → RTA: On arriving at the accident scene, the AMU transmits an accident-scene-arrival message and current conditions of the injured to RTA.

Step 7: RTA → AMU: RTA sends the name of the chosen hospital and the information of the shortest route to the hospital to the AMU.

Step 8: AMU → RTA: On arriving at the hospital, the AMU transmits a hospital-arrival message to RTA to notify the completion of the rescue task.

3.2 The Data Connection Core

In the ATCS, the high security level and the robust key exchange process are, respectively, achieved and developed by using the Data Connection Core (DCC), the format of which is shown in Figure 2. The DCC consists of five parameters, including *AMUID*, e_i, d_i, N_i, and *Cellphone No*, which are stored both in an AMU and RTA when the AMU registers itself with the RTA. *AMUID* is the identity of an AMU and (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. *Cellphone No* is the AMU's cell phone number through which AMU can communicate with RTA.

$$AMUID \mid k_i \mid e_i \mid d_i \mid N_i \mid Cellphone\ No$$

Figure 2 The Format of the DCC

3.3 The OP-Code Table

In the ATCS, the *OP-code* as the first field of a message points out the process and function of the message. With the *OP-code*, both sides of the communication can authenticate whether the message received is really sent by the other side or not. Table 1 lists definitions of the employed OP-codes.

Steps 1 ~ 6 are used by the AMU and RTA during AMU's travel from its current position to the accident scene, while steps 7 ~ 10 are employed by the AMU and RTA when the AMU is on the way to the hospital from the accident scene.

3.4 Parameters and Functions

The parameters and functions utilized by the ATCS are defined as follows.

Table 1 Definitions of Employed OP-Codes

OP-code	Processes and functions
1	Designating the task (by RTA)
2	Replying the designation (by AMU)
3	Directing the AMU along the route (by RTA)
4	Continuously replying RTA with its current location (Accident-scene bound) (byAMU)
5	Continuously directing and monitoring the AMU (Accident-scene bound) (by RTA)
6	Arriving at the accident scene (by AMU)
7	Sending the hospital's address and the shortest route to AMU (by RTA)
8	Continuously replying RTA with its current location (Hospital bound) (by AMU)
9	Continuously directing and monitoring the AMU (Hospital bound) (by RTA)
10	Arriving at the hospital (by AMU)
11 ~ 15	Reserved

3.4.1 The Parameters

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

(1) *AMUID*: the identity of an AMU.

(2) (e_i, d_i, N_i): the individual RSA-triple keys in which e_i is the RSA encryption key, d_i is the RSA decryption key and N_i is a positive integer.

(3) *Cellphone No*: the AMU's Cellphone number.

(4) *OP-code*: the operation code which indicates the process and function of a wireless message.

(5) T_{nonce}: the timestamp of current time.

(6) R_{rj}, $j = 1 \sim 12$: the random numbers generated by the RTA.

(7) R_{aj}, $j = 1 \sim 12$: the random numbers generated by the AMU.

(8) *LA*: the address of the accident scene expressed in longitude and latitude.

(9) *Route*: the route from the AMU's current location to the accident scene.

(10) *RTA-Cellphone-No*: the RTA's cellphone number through which RTA can communicate with the AMU.

(11) DP_i, DK_i, DC_i, $0 \le i \le 18$: dynamic random keys generated by RTA and AMU, independently.

3.4.2 The Functions

The functions employed by the ATCS are defined as follows.

(1) Exclusive-or operator $\oplus$:
Encryption: $c = p \oplus K$,
Decryption: $p = c \oplus K$.

(2) Binary-adder $+_2$:
Encryption: $c = p +_2 K$, where p and K undergo binary

addition, and the carry generated by the addition of the most significant bits is ignored;

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

where $-_2$ denotes the binary subtraction, and $\overline{K}$ is the one's complement of K. When $c \geq K$, it means during the encryption process, no carry is generated in the most-bit addition. The decryption process only reverses the operations of the encryption process. If $c < K$, it implies that in the encryption process, after binary-adding K, the result, i.e., c, is smaller than K, indicating that the carry generated in performing the most-bit addition is omitted. Therefore, in the decryption process, we need to add this carry, i.e., $(K + \overline{K} + 1)$, back to $(c - K) \oplus b$, i.e., $((c - K) + (K + \overline{K} + 1)) = (c + \overline{K} + 1) \oplus b$, to recover the received c.

(3) $RSA\text{-}En(m, e_i)$: An RSA encryption function defined as $RSA - En(m, e_i) = m^{e_i} \bmod N_i$, where m is a plaintext.

(4) $RSA\text{-}De(c, d_i)$: An RSA encryption function defined as $RSA - De(c, d_i) = c^{d_i} \bmod N_i$, where c is a ciphertext.

(5) $En_1(a, b, c)$: An encryption function defined as $En_1(a, b, c) = (a \oplus b) +_2 c$, where a, b, and c are random parameters generated by the ATCS.

(6) $En_2(a, str)$: An encryption function defined as $En_2(a, str) = a \oplus s_1 // a \oplus s_2 // a \oplus s_3 // ... // a \oplus s_n$, where $str = s_1 s_2 s_3 ... s_n$ is a string and "//" denotes concatenation.

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

Example 1: If there is a message, $OP\text{-}code \mid t_{nonce} \mid RSA\text{-}En(R_{r1}, e_i) \mid En_1(R_{r2}, k_i, R_{r1}) \mid En_1(R_{r3}, R_{r1}, R_{r2}) \mid En_1(R_{r4}, R_{r2}, R_{r3}) \mid En_1(R_{r5}, R_{r3}, R_{r4}) \mid En_1(R_{r6}, R_{r4}, R_{r5}) \mid En_2(R_{L1}, LA\ //route) \mid HMAC(R_{r5} \oplus R_{r6})$, transmitted from RTA to an AMU, then $HMAC(R_{r5} \oplus R_{r6})$ is the authentication code generated by invoking a hash function to encrypt the plaintext, $OP\text{-}code \mid t_{nonce} \mid RSA\text{-}En(R_{r1}, e_i) \mid En_1(R_{r2}, k_i, R_{r1}) \mid En_1(R_{r3}, R_{r1}, R_{r2}) \mid En_1(R_{r4}, R_{r2}, R_{r3}) \mid En_1(R_{r5}, R_{r3}, R_{r4}) \mid En_1(R_{r6}, R_{r4}, R_{r5}) \mid En_2(R_{L1}, LA\ //route) \mid HMAC(R_{r5} \oplus R_{r6})$.

(8) $f_1(New\ path)$: An encryption function defined as $f_1(New\ path) = En_2(DC_k \oplus DK_j, New\ path)$.

3.5 The Security Process between AMU and RTA

The security process between an AMU and the RTA as shown in Figure 3 is described below.

Step 1: by RTA

On receiving an AMU-requesting call from an informant U, RTA checks U's nearby AMUs, chooses a suitable one, and requests the AMU to go. When this AMU accepts the task, RTA first retrieves the DCC of the AMU from its DCC database, and stores the DCC in a dynamic record, a record of its dynamic database used to keep track of the rescue task of the AMU. RTA further

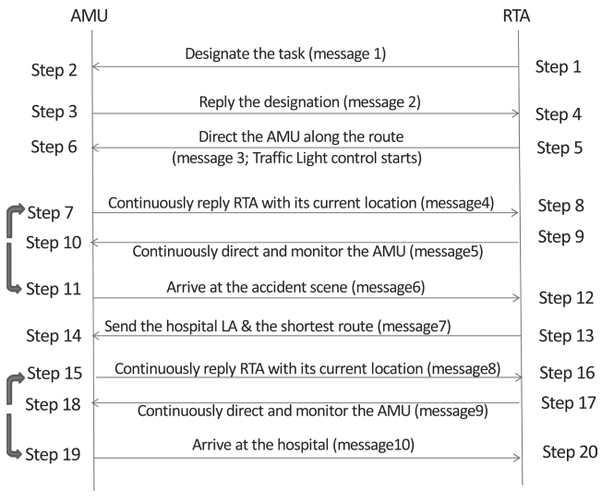

Figure 3 The Proposed Security Process between AMU and RTA

(1) randomly chooses twelve random numbers $R_{r1} \sim R_{r12}$ from its random-number database, and generates a parameter R_{L1} by using four of the chosen random numbers, i.e.,

$$R_{L1} = (R_{r2} +_2 R_{r6}) \oplus (R_{r3} +_2 R_{r5}) \quad (1)$$

(2) generates message 1, the format of which is shown in Figure 4, and then sends the message to the AMU.

(3) randomly chooses twelve random numbers $R_{r1} \sim R_{r12}$ from its random-number database, and generates a parameter R_{L1} by using four of the chosen random numbers, i.e.,

$$R_{L1} = (R_{r2} +_2 R_{r6}) \oplus (R_{r3} +_2 R_{r5}) \quad (1)$$

(4) generates message 1, the format of which is shown in Figure 4, and then sends the message to the AMU.

$$OP\text{-}code \mid t_{nonce} \mid RSA\text{-}En(R_{r1}, e_i) \mid En_1(R_{r2}, k_i, R_{r1}) \mid En_1(R_{r3}, R_{r1}, R_{r2}) \mid En_1(R_{r4}, R_{r2}, R_{r3}) \mid En_1(R_{r5}, R_{r3}, R_{r4}) \mid En_1(R_{r6}, R_{r4}, R_{r5}) \mid En_2(R_{L1}, LA\ //route) \mid HMAC(R_{r5} \oplus R_{r6})$$

Figure 4 The Format of Message 1

(5) randomly chooses twelve random numbers $R_{r1} \sim R_{r12}$ from its random-number database, and generates a parameter R_{L1} by using four of the chosen random numbers, i.e.,

$$R_{L1} = (R_{r2} +_2 R_{r6}) \oplus (R_{r3} +_2 R_{r5}) \quad (1)$$

(6) generates message 1, the format of which is shown in Figure 4, and then sends the message to the AMU.
In this message, OP-code (= 1) is the operation code, t_{nonce} is a timestamp, and $R_{r1} \sim R_{r6}$ are six chosen random numbers. AMU

(7) updates its dynamic record, the format of which is (*AMUID*, k_i, e_i, d_i, N_i, *Cellphone-No*, $R_{r1} \sim R_{r12}$, R_{L1}, *LA*, *route*). In this record, the status of current step is set to 2, and several fields are set to nulls. The values of these fields will be filled in in the following.

Step 2: by AMU

(1) When receiving message 1, AMU verifies whether the *OP-code* of this message meets the status recorded in AMU's dynamic record (= 1), and $t_{received} - t_{nonce}$ is smaller than a pre-defined ΔT or not. If at least one is false, the AMU discards this message and stops this process. Otherwise, it decrypts this message, and

(2) decrypts $RSA\text{-}En(R_{r1}, e_i)$ with d_i where $R_{r1,c} = RAS\text{-}En(R_{r1}, e_i)^{d_i} \bmod N_i$, in which the subscript c is used to discriminate the one calculated by itself from the one retrieved from a received message. Currently, it is message 1.

(3) decrypts $En_1(R_{r2}, k_i, R_{r1})$ by using $R_{r1,c}$ and k_i where $R_{r2,c} = \begin{cases} (y - R_{r1}) \oplus k_i, & if \ \ y \geq R_{r1} \\ (y + R_{r1} + 1) \oplus k_i, & if \ \ y < R_{r1} \end{cases}$, where $y = En_1(R_{r2}, k_i, R_{r1})$.
Only the AMU can decrypt $En_1(R_{r2}, k_i, R_{r1})$ by using $R_{r1,c}$ and k_i since e_i, k_i, d_i, and N_i are only known by the AMU and RTA. The processes of decrypting $R_{r3,c} \sim R_{r6,c}$ are similar to that of decrypting $R_{r2,c}$.

(4) verifies whether $HMAC(R_{r5,c} \oplus R_{r6,c})_c \stackrel{?}{=} HMAC(R_{r5} \oplus R_{r6})_r$ in which the subscript r represents that the $HMAC()$ is retrieved from a received message. If the two expressions are not equal, AMU discards this message and the worker in the AMU calls the RTA to resend message 1.

(5) generates $R_{L1,c}$ by invoking Equation (1).

(6) decrypts $En_2(R_{L1}, LA//route)$ to obtain LA, and the route by using $R_{L1,c}$, i.e., $LA//route = R_{L1,c} \oplus En_2(R_{L1}, LA // route)$.

Step 3: by AMU

In this step, AMU produces twelve random numbers $R_{a1} \sim R_{a12}$, and

(1) sends a message, denoted by message 2, to RTA. The format of this message is shown in Figure 5, in which *OP-code* (= 2) and $R_{a1} \sim R_{a6}$ generated by the AMU are protected by $R_{r1} \sim R_{r6}$ produced by RTA. *CurrentLo* is current location of the AMU expressed also by longitude and latitude.

(2) updates its dynamic record
(*AMUID*, k_i, e_i, d_i, N_i, *Cellphone-No*, $R_{r1} \sim R_{r6}$, $R_{a1} \sim R_{a12}$, R_{L1}, *LA*, *route*, *CurrentLo*) with a part of the data carried in message 2, and *status* is set to 3. Currently, several fields are set to nulls. They will be filled in in the following.

OP-code | *AMUID* | $En_1(R_{a1}, R_{r1}, R_{r2})$ | $En_1(R_{a2}, R_{r2}, R_{r3})$ | $En_1(R_{a3}, R_{r3}, R_{r4})$ | $En_1(R_{a4}, R_{r4}, R_{r5})$ | $En_1(R_{a5}, R_{r5}, R_{r6})$ | $En_1(R_{a6}, R_{r6}, R_{r1})$ | *Reply* | *CurrentLo* | $HMAC(R_{r3} \oplus R_{a6})$

Figure 5 The Format of Message 2

Step 4: by RTA

When receiving message 2, RTA 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 legal one. Otherwise, RTA decrypts this message so as to know the AMU's decision, and starts

(1) decrypting AMU's six encrypted random numbers $R_{a1} \sim R_{a6}$, by using R_{rj}, $1 \leq j \leq 6$, with the following formula.
$R_{aj,c} = \begin{cases} (y - R_{r(j+1)}) \oplus R_{rj}, & if \ \ y \geq R_{r(j+1)} \\ (y + \overline{R_{r(j+1)}} + 1) \oplus R_{rj}, & if \ \ y < R_{r(j+1)} \end{cases}$, $1 \leq j \leq 6$,
where $y = En_1(R_{aj}, R_{rj}, R_{r(j+1)})$.

(2) verifying message 2 by checking to see whether or not $HMAC(R_{r3} \oplus R_{a6,c})_c$ $HMAC(R_{r3} \oplus R_{a6})_r$. If not, RTA discards this message and waits for a legal one. Otherwise, it calls AMU to make sure the accuracy of the content of message 2 through *AMU-Cellphone-No*. *Reply* field in message 2 has three possible values, 1 ~ 3. If the value is 1 (or 2), meaning that AMU can go to the accident scene immediately (in a few minutes), the process goes to step 5. When it is 3, indicating that due to some reasons AMU cannot go, then the process goes back to step 1 to look for another available AMU.

Step 5: by RTA

RTA chooses a specific cellphone number, denoted by *RTA-Cellphone-No*, and sends it to the AMU for urgent needs. When an event unexpectedly occurs, the AMU can call RTA through the cellphone. RTA performs this step by

(1) first generating a parameter R_{L2} to protect *RTA-Cellphone-No*, where $R_{L2} = (R_{a1} \oplus R_{r11}) \oplus (R_{a5} +_2 R_{r10})$.

(2) sending a message, denoted by message 3, to AMU. The format of this message is illustrated in Figure 6, in which *OP-code* = 3 and the six RTA random numbers $R_{r7} \sim R_{r12}$ are encrypted by using the six AMU random numbers $R_{a7} \sim R_{a12}$.

OP-code | $En_1(R_{r7}, R_{a1}, R_{a2})$ | $En_1(R_{r8}, R_{a2}, R_{a3})$ | $En_1(R_{r9}, R_{a3}, R_{a4})$ | $En_1(R_{r10}, R_{a4}, R_{a5})$ | $En_1(R_{r11}, R_{a5}, R_{a6})$ | $En_1(R_{r12}, R_{a6}, R_{a1})$ | $En_2(R_{L2}$, RTA-*Cellphone-No*) | $HMAC(R_{r12} \oplus R_{a6})$

Figure 6 The Format of Message 3

(3) generating dynamic random numbers $DP_0 \sim DP_{18}$, $DK_0 \sim DK_{18}$, and $DC_0 \sim DC_{18}$ to protect traffic light numbers where $DP_j = R_{rj}$, $1 \le j \le 12$; $DP_{j+12} = R_{aj}$, $1 \le j \le 6$; $DK_0 = R_{L1} \oplus R_{L2}$; $DC_0 = R_{L1} +_2 R_{L2}$; $DP_0 = DK_0 +_2 R_{L2}$, $DK_i = [(DP_i \oplus DK_{i-1}) +_2 (R_{L1} \oplus DC_{i-1})] \oplus (R_{L2} \odot DC_{i-1})$, $DC_i = [(DP_i \oplus DK_{i-1}) +_2 (R_{L2} \oplus DC_{i-1})] \oplus (R_{L1} \odot DK_{i-1})$, $1 \le i \le 18$.

(4) updating the AMU's dynamic record, i.e., (*AMUID*, k_i, e_i, d_i, N_i, *Cellphone-No*, *status* (= 4), $R_{r1} \sim R_{r12}$, $R_{a1} \sim R_{a6}$, R_{L1}, R_{L2}, *LA*, *route*, *CurrentLo*, *RTA-Cellphone-No*, $DP_0 \sim DP_{18}$, $DK_0 \sim DK_{18}$, $DC_0 \sim DC_{18}$) with the parameter values newly produced.

Step 6: by AMU

(1) When receiving message 3, AMU checks to see whether or not the *OP-code* of this message meets the status (= 3) recorded in its dynamic record. If not, it discards this message and waits for a legal one. Otherwise, AMU decrypts the encrypted $R_{r7} \sim R_{r12}$ carried in message 3 where $R_{rj,c} = \begin{cases} (y - R_{a(j-5)}) \oplus R_{a(j-6)}, & \text{if } y \ge R_{a(j-5)} \\ (y + \overline{R_{a(j-5)}} + 1) \oplus R_{a(j-6)}, & \text{if } y < R_{a(j-5)} \end{cases}$, $7 \le j \le 12$, where $y = En_1(R_{rj}, R_{a(j-6)}, R_{a(j-5)})$ and $R_{a7} = R_{a1}$. The latter means the initial value of R_{a7} is the same as the value of R_{a1}.

(2) generates $R_{L2,c}$ where $R_{L2,c} = (R_{a1} \oplus R_{r11,c}) \oplus (R_{a5} +_2 R_{r10,c})$.

(3) verifies message 3 by checking to see whether $HMAC(R_{r12,c} \oplus R_{a6})_c \stackrel{?}{=} HMAC(R_{r12} \oplus R_{a6})_r$. If not, it discards this message, calls RTA to resend message 3, and repeats step 6. Otherwise, it

(4) decrypts the encrypted *RTA-Cellphone-No* by using $R_{L2,c}$ where $RTA\text{-}Cellphone\text{-}No_c = R_{L2,c} \oplus En_2(R_{L2,}\ RTA\text{-}Cellphone\text{-}No)_r$.

(5) generates $DP_0 \sim DP_{18}$, $DK_0 \sim DK_{18}$ and $DC_0 \sim DC_{18}$ by invoking $R_{r7,c} \sim R_{r12,c}$ and $R_{L2,c}$ (see substep (3) of step5).

(6) sets counter $i = 0$.

Step 7: by AMU

(1) sets $i = i+1$ and *OP-code* = 4, AMU periodically sends its *CurrentLo* carried in message 4 to RTA until arriving at the accident scene. The format of message 4 is illustrated in Figure 7, in which AMU's *CurrentLo* is protected by DP_j and DK_k, i counts the number of times that AMU sends its current location to RTA, where $j = i$ mod 18+1, $k = i$ mod 19. Next, AMU.

(2) updates its dynamic record, i.e., (*AMUID*, k_i, e_i, d_i, N_i, *Cellphone-No*, status, $R_{r1} \sim R_{r12}$, $R_{a1} \sim R_{a12}$, R_{L1}, R_{L2}, *LA*, *route*, *CurrentLo*, *RTA-Cellphone-No*, $DP_0 \sim DP_{18}$, $DK_0 \sim DK_{18}$, $DC_0 \sim DC_{18}$), with the new information carried in message 4, and *status* is set to 5.

OP-code | *AMUID* | *i* | $En_2(DK_j, DC_k)$ | En_1(*CurrentLo*, DP_j, DK_k) | $HMAC(DP_k \oplus DC_j)$

Figure 7 The Format of Message 4

Step 8: by RTA

When receiving message 4, RTA

(1) verifies whether the *OP-code* of message 4 meets the status (= 4) recorded in the AMU's corresponding dynamic record or not. If not, RTA discards this message and waits for a legal message 4. Otherwise, it

(2) verifies whether $En_2(DK_j, DC_k)_c \stackrel{?}{=} En_2(DK_j, DC_k)_r$. $j = i$ mod 18+1, $k = i$ mod 19. If yes, flag1 = true; Otherwise, flag1 = false.

(3) verifies whether $HMAC(DP_k \oplus DC_j)_r \stackrel{?}{=} HMAC(DP_k \oplus DC_j)_c$. If yes, flag2 = true; Otherwise, flag2 = false. If both flags 1 and 2 are false, due to poor communication quality or receiving a falsified message, RTA discards this message and calls AMU to retransmit message 4. Otherwise, RTA

(4) decrypts the encrypted current location of the AMU from the received message where $CurrentLoc = \begin{cases} (y - DK_k) \oplus DP_j, & \text{if } y \ge DK_k \\ (y + \overline{DK_k} + 1) \oplus DP_j, & \text{if } y < DK_k \end{cases}$, where $y = En_1$ (*CurrentLo*, DP_j, DK_k).

(5) controls traffic lights in front of the AMU on the route immediately.

Step 9: by RTA

(1) RTA sends message 5 to AMU. The format of this message is shown in Figure 8, in which $j = i$ mod 18+1, $k = i$ mod 19, *OP-code* = 5, and *TL-Name* is the name of the next traffic light that should be turned to green. If the value of *Reply* field in message 5 is 1, implying that no new route is required, then f_1(*New path*) is set to Null. If the value is 2, implying that the path has to be changed, then $f_1(New\ path) = En_2(DC_k \oplus DK_j, New\ path)$, and RTA needs to call and warn AMU to follow the new route. After that, RTA's status is set to 6.

OP-code | i | $En_2(DC_j, DK_k)$ | *Reply* | f_1(*New path*) | $En_2(DP_j \oplus DK_k$, *TL-Name*) | $HMAC(DP_j \oplus DC_k)$

Figure 8 The Format of Message 5

(2) RTA updates the AMU's dynamic record, i.e., (*AMUID*, k_i, e_i, d_i, N_i, *Cellphone-No*, *status*, $R_{r1} \sim R_{r12}$, $R_{a1} \sim R_{a6}$, R_{L1}, R_{L2}, *LA*, *route*, *CurrentLo*, *RTA-Cellphone-No*, $DP_0 \sim DP_{18}$, $DK_0 \sim DK_{18}$, $DC_0 \sim DC_{18}$, *TL-Name*), with the new information carried in message 5 and *status* is set to 4 or 6, where 4 and 6 indicate that the guidance is no longer required and is required, respectively.

Step 10: by AMU

On receiving message 5 from RTA, AMU

(1) verifies whether the *OP-code* carried in this message meets the *status* (= 5) recorded in its dynamic record or not. If not, AMU discards this message and waits for a legal one. Otherwise, it

(2) checks to see whether $En_2(DC_j, DK_k)_r \stackrel{?}{=} En_2(DC_j, DK_k)_c$, $j = i \bmod 18+1$, $k = i \bmod 19$. If yes, flag1 = true; Otherwise, flag1 = false.

(3) verifies message 5 by checking to see whether $HMAC(DP_j \oplus DC_k)_r \stackrel{?}{=} HMAC(DP_j \oplus DC_k)_c$. If yes, flag2 = true; Otherwise, flag2 = flase. If both flags 1 and 2 are false, AMU discards this message and calls RTA to enquire the details of message 5. Otherwise, AMU checks the value of *Reply* field conveyed in message 5. If it is 1, then go to step 10-(4). If the value is 2, implying that a new route is given, then it decrypts the encrypted New path, i.e., *New path* = $(DC_k \oplus DK_j) \oplus f_1$(*New path*), where $j = i \bmod 18+1$, $k = i \bmod 19$.

(4) decrypts *TL-Name* where *TL-Name* = $(DP_j \oplus DK_k) \oplus En_2(DP_j \oplus DK_k$, *TL-Name*)

(5) checks current location, if *CurrentLo* ≠ *LA*, meaning that it is now still on its way to the accident scene, AMU updates its dynamic record with parameter values newly generated. *Status* is set to 5 and the process goes to step 7. Otherwise, indicating AMU has arrived at the accident scene, then it

(6) updates its dynamic record with the new information carried in message 5 and goes to step 11.

Step 11: by AMU

(1) On arriving at the accident scene, AMU sends message 6 to RTA. The format of this message is shown in Figure 9, in which *OP-code* is 6 and $R_{a7} \sim R_{a12}$ are protected by e_i, k_i, and recursively by $R_{a9} \sim R_{a11}$. Also, it requires a couple of minutes to move the injured into the AMU. AMU then updates its arguments with new values for the trip from the accident scene to the hospital. Then, AMU

(2) generates new R_{L1} and new R_{L2}, denoted by R'_{L1} and R'_{L2}, respectively, where $R'_{L1} = (R_{r1} +_2 R_{a7}) \oplus (R_{a8} +_2 R_{a9})$ and $R'_{L2} = (Rr_2 +_2 R_{a10}) \oplus (R_{a11} +_2 R_{a12})$.

(3) updates its dynamic record with the new information carried in message 6 where *status* is set to 7.

OP-code | *AMUID* | *RSA-En*(R_{a7}, e_i) | $En_1(R_{a8}, k_i, R_{a7})$ | $En_1(R_{a9}, R_{a7}, R_{a8})$ | $En_1(R_{a10}, R_{a8}, R_{a9})$ | $En_1(R_{a11}, R_{a9}, R_{a10})$ | $En_1(R_{a12}, R_{a10}, R_{a11})$ | $HMAC(R_{a9} \oplus R_{a12})$

Figure 9 The Format of Message 6

Step 12: by RTA

Upon receiving message 6, RTA

(1) verifies whether the *OP-code* carried in message 6 is the same as the *status* (= 6) recorded in this AMU's dynamic record or not. If not, RTA discards this message, and waits for a legal one. Otherwise, it

(2) decrypts the encrypted R_{a7} where the calculated R_{a7} is $R_{a7,c} = (RSA - En(R_{a7}, e_i))^{d_i} \bmod N_i$.

(3) decrypts the encrypted R_{a8} by using R_{a7},c and k_i where $R_{a8,c} = \begin{cases} (y - R_{a7,c}) \oplus k_i, & if \ \ y \geq R_{a7,c} \\ (y + \overline{R_{a7,c}} + 1) \oplus k_i, & if \ \ y < R_{a7,c} \end{cases}$, where $y = En_1(R_{a8}, k_i, R_{a7})$.

(4) decrypts the encrypted $R_{a9} \sim R_{a12}$ where $R_{aj,c} = \begin{cases} (y - R_{a(j-1),c}) \oplus R_{a(j-2),c}, & if \ \ y \geq R_{a(j-1),c} \\ (y + \overline{R_{a(j-1),c}} + 1) \oplus R_{a(j-2),c}, & if \ \ y < R_{a(j-1),c} \end{cases}$, where $y = En_1(R_{aj}, R_{a(j-2)}, R_{a(j-1)})$, $9 \leq j \leq 12$.

(5) verifies whether $HMAC(R_{a9,c} \oplus R_{a12,c})_c \stackrel{?}{=} HMAC(R_{a9} \oplus R_{a12})_r$. If not, AMU discards this message and calls RTA to enquire the details of message 6. Otherwise, RTA

(6) generates new R'_{L1} and new R'_{L2} where $R'_{L1,c} = (R_{r1} +_2 R_{a7,c}) \oplus (R_{a8,c} +_2 R_{a9,c})$ and $R'_{L2,c} = (R_{r2} +_2 R_{a10,c}) \oplus (R_{a11,c} +_2 R_{a12,c})$.

Step 13: by RTA

(1) In this step, RTA sends message 7, which carries the name, phone number, location and address of the designate hospital, to AMU. The format of message 7 is shown in Figure 10, where *OP-code* is 7. RTA then

(2) generates new $DP_0 \sim DP_{18}$, $DK_0 \sim DK_{18}$ and $DC_0 \sim DC_{18}$, which are calculated by using AMU's random numbers $R_{a7} \sim R_{a12}$, old $DP_0 \sim DP_{18}$, old $DK_0 \sim DK_{18}$ and old $DC_0 \sim DC_{18}$, i.e., $DP_{i,c} = (DP_i \oplus R_{a7}) +_2 (R_{a8} \oplus DK_i)$, $DK_{i,c} = (DK_i \oplus R_{a9}) +_2 (R_{a10} \oplus DC_i)$, $DC_{i,c} = (DC_i \oplus R_{a11}) +_2 (R_{a12} \oplus DP_i)$, $0 \leq i \leq 18$.

(3) updates this AMU's dynamic record with the parameter values conveyed in message 7, in which status is set to 8.

OP-code | $En_2(R'_{L1}$, *hospital name//phone number//location and address*$)$ | $En_2(R'_{L2}$, *route*$)$ | $HMAC(R_{a10} \oplus R_{a11})$

Figure 10 The Format of Message 7

Step 14: by AMU

(1) When transporting the injured toward the hospital and receiving message 7, AMU verifies the message by checking to see whether the *OP-code* carried in this message is equal to the status (= 7) kept in its dynamic record, and verifies whether $HMAC(R_{a10} \oplus R_{a11})_r \stackrel{?}{=} HMAC(R_{a10} \oplus R_{a11})_{inside}$. If the message cannot pass both verifications, AMU discards this message and calls RTA to enquire the details of message 7. Otherwise, AMU

(2) decrypts the encrypted hospital information by employing R'_{L1} and $En_2()$ function, i.e., *hospital name// phone number//location* and *address* = $R'_{L1} \oplus En_2(R'_{L1}$, *hospital name//phone number//location and address*$)$.

(3) retrieves the route, and follows the route to go to the hospital where *route* = $R'_{L2} \oplus En_2(R'_{L2}$, *route*$)$.

(4) generates new $DP_0 \sim DP_{18}$, $DK_0 \sim DK_{18}$ and $DC_0 \sim DC_{18}$, all of which are calculated by using AMU's new random numbers $Ra_7 \sim Ra_{12}$, old $DP_0 \sim DP_{18}$, $DK_0 \sim DK_{18}$ and $DC_0 \sim DC_{18}$, i.e., $DP_{i,c} = (DP_i \oplus R_{a7}) +_2 (R_{a8} \oplus DK_i)$, $DK_{i,c} = (DK_i \oplus R_{a9}) +_2 (R_{a10} \oplus DC_i)$, $DC_{i,c} = (DC_i \oplus R_{a11}) +_2 (R_{a12} \oplus DP_i)$, $0 \le i \le 18$.

(5) updates its dynamic record with the new parameter values. After that, Route, $DP_0 \sim DP_{18}$, $DK_0 \sim DK_{18}$ and $DC_0 \sim DC_{18}$ are all new values, and *OP-code* is set to 8.

(6) resets i to 0.

Step 15: by AMU

(1) AMU periodically sends message 8, which carries AMU's current location, i.e., *CurrentLo*, to RTA until it arrives at the hospital. The format is this message is shown in Figure 11, where *OP-code* is 8.

$OP\text{-}code \mid AMUID \mid i \mid En_2(DK_j, DC_k) \mid En_1(CurrentLo, DP_j, DK_k) \mid Speed \mid HMAC(DP_k \oplus DC_j)$

Figure 11 The Format of Message 8, which Is Sent to RTA by AMU

CurrentLo is protected by DP_j and DKk, and i represents the ith time that AMU sends its current location to RTA, where $j = i$ mod 18+1 , $k = i$ mod 19.

(2) updates its dynamic record with the new information carried in message 8 in which *status* is set to 9.

Step 16: by RTA

When receiving current location of AMU, RTA immediately controls traffic lights to support AMU's driving on the route. RTA

(1) first verifies whether the *OP-code* of message 8 is the same as the *status* (= 8) recorded in AMU dynamic record or not. If they are not equal, RTA discards this message and waits for a legal message 8. Otherwise, it

(2) verifies whether $En_2(DK_j, DC_k)_c \stackrel{?}{=} En_2(DK_j, DC_k)_r$ where the subscript c (r) indicates the value of the parameter is obtained by calculation (retrieved from message 8). If they are equal, flag1 = true. Otherwise, flag1 = false.

(3) verifies whether $HMAC(DP_k \oplus DC_j)_r \stackrel{?}{=} HMAC(DP_k \oplus DC_j)_c$. If yes, flag2 = true. Otherwise, flag2 = false.

(4) If both the two flags are false, RTA discards this message and calls AMU to retransmit message 8. Otherwise, RTA

(5) decrypts the encrypted current location of m AMU fro message 8, i.e., *CurrentLo* = $\begin{cases} (y - DK_k) \oplus DP_j, & if\ y \ge DK_k \\ (y + DK_k + 1) \oplus DP_j, & if\ y < DK_k \end{cases}$, where $y = En_1(CurrentLo, DP_j, DK_k)$.

(6) starts controlling those traffic lights on the route from the AMU's current location to the hospital.

Step 17: by RTA

(1) RTA sends message 9 to AMU. The format of this message is illustrated in Figure 12, in which $j = i$ mod 18+1, $k = i$ mod 19, *OP-code* is 9, and *TL-Name* is the name of the next traffic light that has to be turned to green.

$OP\text{-}code \mid i \mid En_2(DC_j, DK_k) \mid Reply \mid f_1(New\ path) \mid En_2(DP_j \oplus DK_k, TL\text{-}Name) \mid HMAC(DP_j \oplus DC_k)$

Figure 12 The Format of Message 9, which Is Sent to AMU by RTA

(2) If the value of the *Reply* field carried in message 9 is 1, implying that the route is still fine, $f_1(New\ path)$ is set to Null. If the value is 2, implying that a new path is required, $f_1(New\ path) = En_2(DC_k \oplus DK_j, New\ path)$, and status is set to 8 or 10, where 8 and 10 indicate that the guidance is still required and no longer required, respectively.

(3) updates this AMU's dynamic record with the parameter values conveyed in message 9.

Step 18: by AMU

On receiving message 9, AMU

(1) verifies message 9 received from RTA by checking to see whether its *OP-code* is the same as the status (= 9) recorded in its dynamic record. If not, AMU discards this message and waits for a legal message 9. Otherwise, it

(2) checks to see whether $En_2(DC_j, DK_k)_r \stackrel{?}{=} En_2(DC_j, DK_k)_c$. If yes, then flag1 = true. Otherwise, flag1 = false.

(3) verifies whether $HMAC(DP_j \oplus DC_k)_r \stackrel{?}{=} HMAC(DP_j \oplus DC_k)_c$. If yes then flag2 = true. Otherwise, flag2 = false. If both the two flags are false, AMU discards this message and calls RTA to enquire the details of message 9. Otherwise, the process continues.

(4) If the value of the parameter *Reply* field is 1, meaning that AMU does not need a new route, the process goes to next substep, i.e., substep 18-(5). If the value of is 2, implying that a new route is required, then it decrypts the encrypted New path, i.e., *New path* = $(DC_k \oplus DK_j) \oplus f_1(New\ path)$, where $j = i$ mod 18+1, $k = i$ mod 19.

(5) AMU decrypts *TL-Name* where *TL-Name* = $(DP_j \oplus DK_k) \oplus En_2(DP_j \oplus DK_k, RG\text{-}Name)$.

(6) updates its dynamic record with the new information carried in message 9. Note that if the value of *Reply* is 2, the route has been substituted by a new one.

(7) AMU checks current location, if $CurrentLo \neq HospitalLo$, meaning that AMU is now still on the way to the hospital, AMU updates its dynamic record with the parameter values newly generated, *status* is set to

9 and the process goes to step15. Otherwise, implying that AMU arrives at the designate hospital, the process goes to step 19.

Step 19: by AMU

(1) On arriving at the hospital, AMU sends message 10 to RTA to inform RTA of the arrival. The format of Message10 is shown in Figure 13, in which *OP-code* is 10.

$OP\text{-}code \mid AMUID \mid En_1(R_{a12}, R_{r10}, R_{r11}) \mid HMAC(R_{r10} \oplus R_{a12})$

Figure 13 The Format of Message 10

(2) AMU updates its dynamic record, (*AMUID*, k_i, e_i, d_i, N_i, *Cellphone-No*, *status*, $R_{r1} \sim R_{r12}$, $R_{a1} \sim R_{a12}$, R'_{L1}, R'_{L2}, *LA*, *route*, *HospitalLo*, *hospital name*, *hospital phone number*, *CurrentLo*, *RTA-Cellphone-No*, *TL-Name*, $DP_0 \sim DP_{18}$, $DK_0 \sim DK_{18}$, $DC_0 \sim DC_{18}$), with the new information, and *status* is set to 1. At last, AMU stores all the information of the dynamic record is its own dynamic database.

Step 20: by RTA

(1) On receiving message 10, RTA verifies whether the *OP-code* carried in this message is the same as the *status* (= 10) kept in AMU dynamic record or not. If not, AMU discards this message, calls RTA to enquire the details of the message and waits for a legal message 10. Otherwise,

(2) RTA further verifies whether $En_1(R_{a12}, R_{r10}, R_{r11})_r \stackrel{?}{=} En_1(R_{a12}, R_{r10}, R_{r11})_c$. If yes, flag1 = true. Otherwise, flag1 = false. Also, RTA continues verifying $HMAC(R_{r10,c} \oplus R_{a12,c})_c \stackrel{?}{=} HMAC(R_{r10} \oplus R_{a12})_r$. If yes, flag2 = true. Otherwise, flag2 = false.

(3) If both the two flags are false, illustrating that it is an error message generated by hackers or the transmission message is seriously interfered, RTA calls AMU to enquire the content of message 10, and the process goes to step 19. Otherwise, implying that AMU has arrived at the hospital, RTA calls AMU to confirm the arrival.

RTA updates AMU's dynamic record, (*AMUID*, k_i, e_i, d_i, N_i, *Cellphone-No*, *status*, $R_{r1} \sim R_{r12}$, $R_{a1} \sim R_{a12}$, R'_{L1}, R'_{L2}, *LA*, *route*, *hospital LA*, *hospital name*, *hospital phone number*, *CurrentLo*, *RTA-Cellphone-No*, *TL-Name*, $DP_0 \sim DP_{18}$, $DK_0 \sim DK_{18}$, $DC_0 \sim DC_{18}$), with the new information. At last, RTA stores all information in its own database, and the rescue task is finished.

4 Security Analyses

In this section, we analyze the security of (1) the key exchange process, i.e., steps 1 ~ 6 and steps 11 ~ 12; (2) the transmitted data which is protected by $En_2()$ function; (3) the wireless messages; (4) the double authentication mechanism. We also describe how the ATCS effectively defends four common attacks, including eavesdropping attack, forgery attack, replay attack, and man-in-the-middle attack.

4.1 Security of the Key Exchange Process

Two operators, i.e., exclusive-or $\oplus$ and binary-adder $+_2$, are employed by the ATCS. Let X and Y be two keys, each of which is n bits in length. The probability p of recovering the value of (X, Y) from illegally intercepted $X \oplus Y$ ($X +_2 Y$) on one trial is $P = \frac{1}{2^n}$ [22]. What is the security level of random numbers $R_{r1} \sim R_{r12}$, $R_{a1} \sim R_{a12}$ when they are transmitted between AMU and RTA.

Lemma 1: Assume that the random number R_r as a key is n-bits in length. The probability p of recovering the value of R_r from illegally intercepted $RSA\text{-}En(R_r, e_i)$ on one trial is $P = \frac{1}{2^n}$.

Proof: $\because RSA\text{-}En(R_r, e_i) = R_r^{e_i} \bmod N_i$. However, the RSA-triple keys (e_i, d_i, N_i) of an AMU are only known by the AMU and RTA before the wireless communication begins. Hence, hackers cannot obtain the RSA-triple keys (e_i, d_i, N_i) from the messages delivered through the wireless channels.

Moreover, since different AMUs are given different (e_i, d_i, N_i)s, hackers cannot acquire information concerning the (e_i, d_i, N_i) from other AMU's wirelessly delivered messages. The lack of the values of e_i, d_i, and N_i makes hackers unable to break $RSA\text{-}En(R_r, e_i)$ to obtain R_r. The only possible method to obtain R_r is by blind guessing. Hence, the probability p of recovering the value of R_r from illegally intercepted $RSA\text{-}En(R_r, e_i)$ on one trial is $P = \frac{1}{2^n}$.

In steps 1 ~ 6 and steps 11 ~ 12, the transmitted random numbers R_{rj}, $2 \le j \le 12$, and R_{aj}, $1 \le j \ne 7 \le 12$, are protected by a security scheme, called the keys-protection-key chain mechanism. Since some keys are known only by the AMU and RTA before the wireless communication starts (we call them connection keys), a transmitted key can be well protected by encrypting it with the connection keys. In fact, the keys-protection-key mechanism is a protection chain, in which the first protected transmitted key is used to encrypt/protect the second transmitted key, which together with the first transmitted key is then employed to encrypt the third transmitted key, and so on.

Lemma 2: The transmitted random numbers R_{rj}, $2 \le j \le 12$, and R_{aj}, $1 \le j \ne 7 \le 12$, employed by the ATCS are protected by a keys-protection-key chain mechanism. Let each of the transmitted random numbers r and keys x and y be n-bits in length. The probability p of recovering the value of r from illegally intercepted $En_1(r, x, y)$ on one trial is $p = \frac{1}{2^n}$.

Proof: First: ∵ $En_1(r, x, y) = (r \oplus x) +_2 y$, Decrypt it, then

$$r = \begin{cases} (En_1(r,x,y) - y) \oplus x, & if \;\; En_1(r,x,y) \geq y \\ (En_1(r,x,y) + \overline{y} + 1) \oplus x, & if \;\; En_1(r,x,y) < y \end{cases} \quad (2)$$

Equation (2) shows that if hackers wish to acquire the exact value of r from the illegally intercepted $En_1(r, x, y)$, they need the exact values of x and y. However, x and y are only known by AMU and RTA, hackers do not know their values. Hence, the probability p of recovering r, x and y from $En_1(r, x, y)$ by invoking Equation (2) is $\left(\frac{1}{2^n}\right)^2$ which is very smaller than $\frac{1}{2^n}$, the probability of blind guessing the value of r on one trial when $En_1(r, x, y)$ is known, showing that, no matter whether Equation (2) is employed or not, the probability p of recovering the value of r from a known $En_1(r, x, y)$ is $p = \frac{1}{2^n}$.

Second: Message 1 and Lemma 1 indicate that the transmitted random number R_{r1} is well protected by employing $RSA\text{-}En(R_{r1}, e_i)$. The key R_{r1} as a connection key is only known by the AMU and RTA. In message 1, R_{r1} together with the individual characteristic key k_i are used to protect the transmitted random number R_{r2}. Then R_{r2} is well protected by employing $En_1(R_{r2}, k_i, R_{r1})$, and the keys R_{r1} and R_{r2} are now new connection keys which are only known by the AMU and RTA. They in message 1 are used to protect the transmitted random number R_{r3} by employing $En_1(R_{r3}, R_{r1}, R_{r2})$, and R_{r4}, R_{r5}, and R_{r6} are each protected by the similar method.

Hence, the transmitted random numbers R_{rj}, $2 \leq j \leq 6$, in message 1 are protected by a keys-protection-key chain mechanism. Similarly, the transmitted random numbers R_{rj}, $7 \leq j \leq 12$, and R_{aj}, $1 \leq j \neq 7 \leq 12$, appearing in messages 2, 3 and 6, are protected by a keys-protection-key chain mechanism. Q.E.D.

Lemma 3: In message 1, $HMAC(R_{r5} \oplus R_{r6})$ is an authentication code with three security functions, including authentication, non-repudiation and integrity.

Proof:

(Proof of authentication)

Lemma 1 and Lemma 2 show that the transmitted random numbers R_{rj}, $1 \leq j \leq 6$, in message 1 were well protected. Hence, the only mechanism that can correctly generate the authentication code $HMAC(R_{r5} \oplus R_{r6})$ should be the one with the DCC of the AMU. However, hackers cannot acquire the correct DCC of the AMU so that they cannot correctly generate $HMAC(R_{r5} \oplus R_{r6})$. Only the legitimate AMU who has the correct DCC can make $HMAC(R_{r5} \oplus R_{r6})_c = HMAC(R_{r5} \oplus R_{r6})_r$ where the subscripts c and r stand for calculation and received, respectively. Those illegitimate hackers who have no DCC of the AMU cannot achieve this.

(Proof of non-repudiation)

From the analysis above, only the legitimate AMU who has the correct DCC can make $HMAC(R_{r5} \oplus R_{r6})_c = HMAC(R_{r5} \oplus R_{r6})_r$. This implies that message 1 is sent by the one who has the correct DCC, indicating that the AMU is a legitimate one.

(Proof of the integrity)

$HMAC(R_{r5} \oplus R_{r6})$ is the authentication code generated by invoking a hash function performed on the plaintext, *OP-code* | t_{nonce} | $RSA\text{-}En(R_{r1}, e_i)$ | $En_1(R_{r2}, k_i, R_{r1})$ | $En_1(R_{r3}, R_{r1}, R_{r2})$ | $En_1(R_{r4}, R_{r2}, R_{r3})$ | $En_1(R_{r5}, R_{r3}, R_{r4})$ | $En_1(R_{r6}, R_{r4}, R_{r5})$ | $En_2(R_{L1}$, *LA//route*), with the key, $R_{r5} \oplus R_{r6}$. If either the plaintext or the key has been illegally tampered, then $HMAC(R_{r5} \oplus R_{r6})_c \neq HMAC(R_{r5} \oplus R_{r6})_r$ since the value of $HMAC(R_{r5} \oplus R_{r6})$ cannot be correctly calculated by those hackers who have no correct DCC of the AMU. Hence, if $HMAC(R_{r5} \oplus R_{r6})_c = HMAC(R_{r5} \oplus R_{r6})_r$, it means that message 1 has not been illegally tampered and the integrity has been maintained. Q.E.D.

4.2 Security of the Data Protected by $En_2()$ Function

$En_2(a, str) = a \oplus s_1 // a \oplus s_2 // a \oplus s_3 // ... // a \oplus s_n$ indicates that *str* is protected by key *a*. But, it is a fixed key encryption mode. Someday *str* may be cracked by Violence Act attacks, even key *a* is unknown by hackers. For message 1, the AMU may arrive at the accident scene before $En_2(R_{L1}$, *LA//route*) is cracked by hackers. However, even R_{L1}, *LA*, and *route* are known by hackers, the ATCS is still secure since they are used only once. In the next rescue task, they will be regenerated and of course are different from those produced in the underlying task. In fact, the two sets of data of two rescue tasks are unrelated. Further, $R_{L1} = (R_{r2} +_2 R_{r6}) \oplus (R_{r3} +_2 R_{r5})$ indicates that R_{r2}, R_{r6}, R_{r3}, and R_{r5} are still secure, even through R_{L1} is known by hackers.

4.3 Security of the Delivered Messages

In message 1, random numbers R_{r5} and R_{r6} are protected by the keys-protection-key mechanism. Hence, $R_{r5} \oplus R_{r6}$ is unknown to hackers. The delivered messages employing $HMAC(K)$ have three security functions, including authentication, non-repudiation and integrity [22]. Furthermore, if the messages delivered between RTA and AMU employ the combination of *OP-code*, t_{nonce} and $HMAC(R_{r5} \oplus R_{r6})$, they can effectively defend the replay attack [22]. Hence, the security levels of the messages delivered in and protected by ATCS are high.

4.4 Security of the Double Authentication Mechanism

In order to have a more secure, flexible, and fault-tolerant authentication mechanism to protect messages wirelessly delivered between AMU and RTA, ATCS adopts a mutual authentication mechanism to transmit messages 4, 5, 8, and 9.

In message 4, both $En_2(DK_j, DC_k)$ and $HMAC(DP_k \oplus DC_j)$ are authentication codes, in which (1) if the two communication parties, i.e., AMU and RTA, have commonly shared dynamic random numbers DK_j, and DC_k, then AMU can correctly produce an authentication code, $En_2(DK_j, DC_k)$, on its side, and RTA can perform authentication on the other side; (2) not only dynamic random numbers DP_k and DC_j should be commonly shared by the two communication parties, i.e., AMU and RTA, but also the whole message of message 4 cannot be altered in the situation where the authentication code, $HMAC(DP_k \oplus DC_j)$, produced on the AMU side can be correctly authenticated by the RTA. Obviously, this authentication mechanism may be affected by the unstable transmission of message 4. For example, if the signal is interfered, the authentication result will be incorrect.

To increase the security level, flexibility, and fault-tolerant capability of the authentication mechanism for the wirelessly delivered messages, we adopt the double authentication mechanism, in which if both authentication codes, $En_2(DK_j, DC_k)$ and $HMAC(DP_k \oplus DC_j)$, pass the authentication, this indicates that the communication is valid and the communication signal is stable. But if only one of the two authentication codes, $En_2(DK_j, DC_k)$ or $HMAC(DP_k \oplus DC_j)$, passes the authentication, the communication is still valid. But the communication signal is unstable. In this case, AMU and RTA can communicate with each other through cell phones to confirm the information transmitted between them. If both authentication codes, $En_2(DK_j, DC_k)$ and $HMAC(DP_k \oplus DC_j)$, fail, that means the delivered message is invalid and the communication quality is poor. In this case, AMU and RTA should contact each other also through cell phones to confirm the information delivered between them.

4.5 Cryptanalysis of Attacks

The ATCS can effectively defend eavesdropping, forgery, replay, and Man-in-the-middle attacks.

4.5.1 Preventing Eavesdropping Attacks

Eavesdropping due to the wireless nature is a type of attack not easy to be discovered. Hackers may maliciously intercept the messages sent by AMU or RTA, and analyze the messages to acquire useful information.

In the ATCS, hackers can only acquire random numbers $R_{r1} \sim R_{r6}$ from the illegally intercepted message 1. However, from Lemma 1 and Lemma 2, we can comprehend that the probability p of recovering the value of R_{r1} from known $RSA\text{-}En(R_{r1}, e_i)$ is $p = \frac{1}{2^n}$, the probability p of recovering the value of R_{r2} from known $En_1(R_{r2}, k_i, R_{r1})$ is also $p = \frac{1}{2^n}$, and the probability p of recovering each of the value of R_{rj}, $3 \le j \le 6$, from known $En_1(R_{rj}, R_{r(j-2)}, R_{r(j-1)})$, $3 \le j \le 6$ is $p = \frac{1}{2^n}$ as well, showing that $R_{r1} \sim R_{r6}$ are well protected. Furthermore, since the encryption key $R_{r5} \oplus R_{r6}$ in $HMAC()$ is unknown by hackers, they cannot produce the correct authentication code $HMAC(R_{r5} \oplus R_{r6})$, implying that hackers cannot easily crack the delivered random numbers and the authentication code, solve the transmitted messages and acquire the plaintext, meaning the plaintext is secure.

4.5.2 Preventing Forgery Attacks

Hackers often masquerade themselves as legitimate AMUs or the RTA to acquire the authentication information. Namely, if a system does not provide a mutual authentication, a hacker may be considered as a legitimate AMU (or RTA), and then the messages sent to the RTA (or AMU) will be treated as legal ones.

Lemma 3 shows that the key exchange mechanism of the ATCS preserves mutual authentication, implying that only the one who has the DCC can correctly generate the dynamic authentication code $HMAC(R_{r5} \oplus R_{r6})$. The forged messages generated by hackers who do not have the DCC cannot pass the authentication and will be discarded by AMU or RTA. That means the ATCS can defend forgery attacks effectively.

4.5.3 Preventing Replay Attacks

When intercepting an authentication message, hackers will tamper it and send it to AMU or RTA to gain the trust. Hackers may also send duplicate messages two or more times to AMU or RTA, to confuse the receiver which messages are the legal ones.

In message 1, both T_{nonce} and $HMAC(R_{r5} \oplus R_{r6})$ provide the security functions which can effectively defend the replay attacks.

If hackers illegally duplicate message 1, and resend it, then T_{nonce} contained in this message is very different from current time so that $T_{received} - T_{nonce} \ge \Delta T$ where ΔT is a predefined short time period. The message will be discarded by the AMU. If hackers modify T_{nonce} to current time, the value of calculated $HMAC(R_{r5} \oplus R_{r6})$ will change, and also without the correct DCC, hackers cannot calculate the correct value of $HMAC(R_{r5} \oplus R_{r6})$. Hence, $HMAC(R_{r5} \oplus R_{r6})_c$ will not be equal to $HMAC(R_{r5} \oplus R_{r6})$r, indicating that the security function provided by T_{nonce} and $HMAC(R_{r5} \oplus R_{r6})$ can effectively defend the replay attacks.

Furthermore, sending the duplicated message 2 to RTA is also useless since the time point of sending the duplicated message 2 is very later than the time point when the original one was delivered. When the RTA receives message 2 from the legitimate AMU, and message 2 passes the authentication test, the internal state of the RTA will be set to the next state. But the state carried in the *OP-code* of the duplicated message 2 remains in its original state, which does not meet the state of the receiver. The other duplicated messages have the similar phenomenon. Hence, the ATCS can effectively defend the replay attack.

4.5.4 Preventing Man-in-the-Middle Attacks

Each message has its own *HMAC*(). If the hackers grab the message and tamper it, the calculated and received *HMAC*()s will be different. Also, even though the hackers grab the message, they cannot decrypt the message because all delivered random numbers are protected by the RSA algorithm. Without the random numbers, i.e., the encryption keys, hackers cannot decrypt the protected parameters.

5 Conclusions and Future Work

In this study, we propose the ATCS, in which when an accident occurs, RTA searches the most suitable AMU, computes the shortest path from the AMU's current position to the accident scene, and controls traffic lights on the path so that the AMU can rush to the accident scene without traffic delay. When the AMU is now on the way to the designate hospital, the RTA does the same.

We use RSA algorithm and keys-protection-key chain mechanism to protect the random numbers delivered through wireless channels. Without decryption keys, hackers cannot decrypt the encrypted parameters. Also, time stamps and *HMAC*() are deployed so that the transmitted messages are well protected to avoid Replay and Man-in-the-middle attacks. The Appendix of this paper summarizes the authentications performed in all steps of the ATCS.

In the future, we would like to develop the proposed system's formal behavior and reliability models so that users can know the behavior and reliability before using it [23]. We also like to change the role of controlling the traffic lights from the RTA to AMU. The reason is that once some exception handling is required by the AMU, e.g., if there is another traffic accident in front of the AMU, then the AMU has to change its path. Now, even though the traffic lights of the original path are under control, the AMU cannot go ahead. If traffic lights are under the AMU's control, the problem can be solved. Those constitute our future research.

Acknowledgements

The work was supported in part by TungHai University under the project GREENs and the National Science Council, Taiwan under Grants NSC 101-2221-E-029-003-MY3, and NSC 100-2221-E-029-018.

References

[1] R. P. Gonzalez, G. R. 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*, *American Journal of Surgery*, Vol.197, No.1, 2009, pp.30-34.

[2] Rocio Sánchez-Mangas, Antonio García-Ferrrer, Aranzazu de Juan and Antonio Martin Arroyo, *The Probability of Death in Road Traffic Accidents. How Important Is a Quick Medical Response?*, *Accident Analysis and Prevention*, Vol.42, No.4, 2010, pp.1048-1056.

[3] *Part 12: From Science to Survival: Strengthening the Chain of Survival in Every Community*, *Resuscitation*, Vol.46, No.1-3, 2000, pp.417-430.

[4] Rade B. Vukmir, *Survival from Pre-hospital Cardiac Arrest Is Critically Dependent upon Response Time*, *Resuscitation*, Vol.69, No.2, 2006, pp.229-334.

[5] Cheng-Siong Lim, Rosbi Mamat and Thomas Bräunl, *Impact of Ambulance Dispatch Policies on Performance of Emergency Medical Services*, *IEEE Transactions on Intelligent Transportation Systems*, Vol.12, No.2, 2011, pp.624-632.

[6] Peter T. Pons and Vincent J. Markovchick, *Eight Minutes or Less: Does the Ambulance Response Time Guideline Impact Trauma Patient Outcome?*, *The Journal of Emergency Medicine*, Vol.23, No.1, 2002, pp.43-48.

[7] John F. Repede and John 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, 1994, pp.567-581.

[8] M. Castrén, R. Karlsten, F. Lippert, E. F. Christensen, E. Bovim, A. M. Kvam, I. Robertson-Steel, J. Overton, T. Kraft, L. Engerstrom and L. G. C. Riego, *Recommended Guidelines for Reporting on Emergency Medical Dispatch When Conducting Research in Emergency Medicine: The Utstein Style*, *Resuscitation*, Vol.79, No.2, 2008, pp.193-197.

[9] A. K. Marsden, *Getting the Right Ambulance to the Right Patient at the Right Time*, *Accident and Emergency Nursing*, Vol.3, No.4, 1995, pp.177-183.

[10] The NHS Information Centre, *Ambulance Services England 2008-2009*, 2009, http://www.hscic.gov.uk/catalogue/PUB00501/ambu-serv-eng-2008-2009-rep.pdf

[11] Michel Gendreau, Gilbert Laporte and Frédéric Semet, *A Dynamic Model and Parallel Tabu Search Heuristic for Real-Time Ambulance Relocation*, *Parallel Computing*, Vol.27, No.12, 2001, pp.1641-1653.

[12] Michael O. Ball and Feng L. Lin, *A Reliability Model Applied to Emergency Service Vehicle Location*, *Operations Research*, Vol.41, No.1, 1993, pp.18-36.

[13] John J. M. Black and Gareth D. Davies, *International EMS Systems: United Kingdom*, *Resuscitation*, Vol.64, No.1, 2005, pp.21-29.

[14] Luce Brotcorne, Gilbert Laporte and Frédéric Semet, *Ambulance Location and Relocation Models*, *European Journal of Operational Research*, Vol.147, No.3, 2003, pp.451-463.
[15] Richard L. Church and Charles S. ReVelle, *The Maximal Covering Location Problem*, *Papers of the Regional Science Association*, Vol.32, No.1, 1974, pp.101-118.
[16] Constantine Toregas, Ralph Swain, Charles ReVelle and Lawrence Bergman, *The Location of Emergency Service Facilities*, *Operations Research*, Vol.19, No.6, 1971, pp.1363-1373.
[17] Michel Gendreau, Gilbert Laporte and Frédéric Semet, *Solving an Ambulance Location Model by Tabu Search*, *Location Science*, Vol.5, No.2, 1997, pp.75-88.
[18] David E. Persse, Craig B. Key, Richard N. Bradley, Charles C. Miller and Atul Dhingra, *Cardiac Arrest Survival as a Function of Ambulance Deployment Strategy in a Large Urban Emergency Medical Services System*, *Resuscitation*, Vol.59, No.1, 2003, pp.97-104.
[19] Graham Nichol, Allan S. Detsky, Ian G. Stiell, Keith O'Rourke, George Wells and Andreas Laupacis, *Effectiveness of Emergency Medical Services for Victims of Out-of-Hospital Cardiac Arrest: A Metaanalysis*, *Annals of Emergency Medicine*, Vol.27, No.6, 1996, pp.700-710.
[20] Mickey S. Eisenberg, Bruce T. Horwood, Richard O. Cummins, Robin Reynolds-Haertle and Thomas R. Hearne, *Cardiac Arrest and Resuscitation: A Tale of 29 Cities*, *Annals of Emergency Medicine*, Vol.19, No.2, 1990, pp.179-186.
[21] Chun-Hsin Cheng, *A Secure Ambulance Communication Protocol for VANET*, Master Thesis, Chaoyang University of Technology, Taichung, Taiwan, 2010.
[22] Yi-Li Huang, Fang-Yie Leu and Ko-Chung We, *A Secure Communication over Wireless Environments by Using a Data Connection Core*, *Mathematical and Computer Modelling*, Vol.58, No.5-6, 2013, pp.1459-1474.
[23] Yu-Hsiu Chuang, Chi-Yuan Chen, Tzong-Chen Wu and Han-Chieh Chao, *Establish a Secure and Trustworthy Ubiquitous ICT Environment for Educational Systems: A Case Study*, *Journal of Intelligent Manufacturing*, Vol.23, No.4, 2012, pp.965-975.

Biographies

Yi-Li Huang received his master degrees from National Central University of Physics, Taiwan, in 1983. His research interests include cryptography, information security, network security, and solar active-tracking system. He is currently a senior instructor of TungHai University, Taiwan, and director of information security laboratory of the University. He is also a member of IEEE Computer Society.

I-Long Lin received his PhD degree in computer and information science from National Taiwan University of Science and Technology in 1997. His research interests include information security management system, hacking & countermeasure, digital evidence & forensic Computing and cybercrimes investigation. He is currently a Professor of Yuanpei University of Medical Technology.

Fang-Yie Leu received his PhD degree from National Taiwan University of Science and Technology, Taiwan, in 1991. His research interests include wireless communication, network security, Grid applications and Security. He is currently a professor and the chairperson of Information Management Department, TungHai University, Taiwan, and one of the editorial board members of at least 7 journals.

Jung-Chun Liu received his PhD degree from the Department of Electrical and Computer Engineering at the University of Texas at Austin in 2004. He is currently an assistant professor in the Computer Science Department, TungHai University, Taiwan. His research interests include cloud computing, embedded systems, wireless networking, network security, artificial intelligence, and wireless sensor networks.

Fuu-Cheng Jiang was the recipient of the Best Paper Award at the 5th International Conference on Future Information Technology 2010 (FutureTech 2010), which ranked his paper first among the 201 submittals. He has served dozens of the TPC for worldwide international conferences. His research interests include network modeling, cloud computing, wireless networks and simulation.

William Cheng-Chung Chu, a professor of the Department of Computer Science. He had served as the Dean of Research and Development Office at TungHai University from 2004 to 2007, and the dean of Engineering College of TungHai University from 2008 to 2011, Taiwan. His current research interests include software engineering, embedded systems, and E-learning.

Chao-Tung Yang received his PhD in Computer Science from National Chiao Tung University in July 1996. He is now a Professor of Computer Science at TungHai University in Taiwan. His present research interests are in cloud computing and service, grid computing, parallel computing, and multicore programming.

Appendix

The summary of the authentications performed in the ATCS.

Figure A1 The Summary of the Verifications in All Steps of the Proposed System

Step #	Sender	Authentication	Verifier
Step 2	RTA	$HMAC(R_{r5,c} \oplus R_{r6,c})_c \stackrel{?}{=} HMAC(R_{r5} \oplus R_{r6})_r$	AMU
Step 4	AMU	$HMAC(R_{r3} \oplus \mathrm{R}_{a6,c})_c \stackrel{?}{=} HMAC(R_{r3} \oplus R_{a6})_r$	RTA
Step 6	RTA	$HMAC(R_{r12,c} \oplus R_{a6})_c \stackrel{?}{=} HMAC(R_{r12} \oplus R_{a6})_r$	AMU
Step 8	AMU	$En_2(DK_j, DC_k)_c \stackrel{?}{=} En_2(DK_j, DC_k)_r$	RTA
Step 8	AMU	$HMAC(DP_k \oplus DC_j)_c \stackrel{?}{=} HMAC(DP_k \oplus DC_j)_r$	RTA
Step 10	RTA	$En_2(DC_j, DK_k)_c \stackrel{?}{=} En_2(DC_j, DK_k)_r$	AMU
Step 10	RTA	$HMAC(DP\mathrm{j} \oplus DC_k)_c \stackrel{?}{=} HMAC(DP_j \oplus DC_k)_r$	AMU
Step 12	AMU	$HMAC(R_{a9,c} \oplus R_{a12,c})_c \stackrel{?}{=} HMAC(R_{a9} \oplus R_{a12})_r$	RTA
Step 14	RTA	$HMAC(R_{a10} \oplus R_{a11})_r \stackrel{?}{=} HMAC(R_{a10} \oplus R_{a11})_c$	AMU
Step 16	AMU	$En_2(DK_j, DC_k)_c \stackrel{?}{=} En_2(DK_j, DC_k)_r$	RTA
Step 16	AMU	$HMAC(DP_k \oplus DC_j)_r \stackrel{?}{=} HMAC(DP_k \oplus DC_j)_c$	RTA
Step 18	RTA	$En_2(DC_j, DK_k)_r \stackrel{?}{=} En_2(DC_j, DK_k)_c$	AMU
Step 18	RTA	$HMAC(DP_j \oplus DC_k)_r \stackrel{?}{=} HMAC(DP_j \oplus DC_k)_c$	AMU
Step 20	AMU	$En_1(R_{a12}, R_{r10}, R_{r11})_r \stackrel{?}{=} En_1(R_{a12}, R_{r10}, R_{r11})_c$	RTA
Step 20	AMU	$HMAC(R_{r10,c} \oplus R_{a12,c})_c \stackrel{?}{=} HMAC(R_{r10} \oplus R_{a12})_r$	RTA