IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 6, 2018, accepted July 11, 2018, date of publication July 18, 2018, date of current version August 15, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2856904*

# Securing Offline Delivery Services by Using Kerberos Authentication

**HUI LI[1], YI NIU[2], JUNKAI YI[1], AND HONGYU LI[1]**
[1]Department of Computer Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China
[2]China National Publications Import and Export (Group) Corporation, Beijing 100020, China

Corresponding author: Hui Li (ray@mail.buct.edu.cn)

This work was supported by the National Natural Science Foundation of China through grants of the general technical foundation research joint fund under Project U1636208.

**ABSTRACT** With the rise of crowdsourcing economies, the home delivery business is now undergoing rapid development. The authentication schemes that are currently used in the home delivery business remain unanalyzed from the provable authenticity perspective. In this paper, we generalize the ''cross-realm authentication path'' of the Kerberos protocol and apply it to the online ordering and offline delivery business. We design a Kerberos-based scheme for the crowdsourcing delivery model to establish authenticity for principals, including customers, suppliers, and deliverymen. Then we extend it to some more complex models, such as the receiving agent model, the unified service of franchised chain model, and the relay delivery model. The authentication schemes can also be incorporated into the door access and guard system to enhance the physical security in smart cities.

**INDEX TERMS** Kerberos, home delivery, authentication path, crowdsourcing, cross-realm, ordering platform.

## I. INTRODUCTION

Worldwide, the business of delivering ordered goods (e.g. foods, books, and flowers) to home is now undergoing rapid development, as new platforms race to capture markets and customers across the Europe, America, and Asia. Take food delivery business of USA for example, consumer spending on food delivery in 2015 was worth around 30 billion dollars - 4 billion dollars of this was accounted for by home delivery sales [1]. A November 2016 survey found that 20 percent of respondents used food delivery at least once a week. As more and more consumers are becoming used to shopping by using apps, steady rapid growth of this market is anticipated for the next five years.

It is somewhat surprising that the schemes that are actually used in working and deployed scheme remain unanalyzed from the provable security perspective. Some security problems that have been ignored before are raised nowadays. For example, a Houston homeowner was robbed and severely beaten after he opened his front door to a man posing as a deliveryman [2]. For another example, many new deliverymen had bitter experiences of unintentionally delivering goods to the wrong address.

The goal of this paper is to design an authentication scheme for the home delivery models. The scheme is expected to establish authenticity for principals including customers, suppliers, and deliverymen.

The authentication requirements in the above-mentioned scenes are similar to that on the Internet indeed. The major difference is that the real-life entities are persons, rather than workstations or PC terminals. We need develop an authentication protocol to address these real-life authentication problems. However, formally establishing a reliable authentication protocol is not an easy task. We recall Kerberos [3], an authentication protocol which has been widely used in the computer network community.

In the late 1980s, MIT first designed Kerberos to protect network services in the Project Athena [4]. Kohl and Neuman [5] published the fifth edition of the Kerberos specification in 1993 to remove some limitations. RFC 4120, obsoleting RFC 1510 in 2005, clarified aspects of the protocol and its intended use [6]. In 2007, MIT made an implementation of Kerberos freely available, under copyright permissions similar to those used for BSD [7].

Butler *et al.* [8] provided the detailed verification for the Kerberos V5 specification. Many researchers also proposed extensions and revisions. For example, Cervesato *et al.* [9] and Rowe *et al.* [10] reported a man-in-the-middle attack on PKINIT, and proposed a public key extension for the

Kerberos V5 protocol. Liu *et al.* [11] developed a binary tree code algorithm to alleviate the Kerberos server bottleneck problem.

Kerberos has been successfully used in Windows as its default authentication method [12]. Many UNIX operating systems, including Fedora, Ubuntu, and FreeBSD, also contain Kerberos authentication tools [13].

We decide to exploit Kerberos in the home delivery business, because Kerberos is technically mature and structurally sound. Moreover, the Kerberos protocol can be designed to operate across organizational boundaries. Namely, a client in one organization can be authenticated to a server in another organization. Many applications such as next generation network application can be found in literature [14] [15]. This cross-realm Kerberos may facilitate the authentication design for some complicated network applications.

However, direct imitation of the typical cross-realm network authentication is not applicable for the home delivery business. Therefore, we proposed a generalized authentication path for cross-realm Kerberos. With the aid of the new idea, we can obtain the authentication solutions for the crowdsourcing model. Furthermore, the method can be applied to more sophisticated delivery models without much difficulty. We have tested the method in a book ordering and home delivery platform. The platform has run for couples of months with no negative feedback.

The main contributions of this paper include:

1) This is the first work to generalize the authentication path of Kerberos. The generalized authentication path endows the cross-realm Kerberos applications with more flexibility.
2) It is also the first work to apply Kerberos to the offline real-life business. In the past, people were used to moving the real-life business to the Internet. It seems unusual to apply a mature network protocol to the offline scenario in reverse.
3) The proposed scheme can provide strong cryptographic authentication for the fast growing crowdsourcing business.

The reminder of this paper is organized as follows. In Section II, we discuss the security requirements of home delivery models and give a background of the Kerberos protocol. In Section III, we analyze the similarity and difference between the offline business scenario and the online Kerberos environment. We propose the approach and the algorithm to construct the generalized cross-realm authentication path, and then apply it to the home delivery business. Section IV deals with applications of the extended Kerberos to more complicated home delivery models. Section V concludes the paper finally.

## II. HOME DELIVERY APPLICATION OF KERBEROS

We start with a typical scenario of the food ordering and home delivery platform. This platform is similar to other ordering and home delivery platforms such as flower and book retail platforms. We analyze its authentication requirements and then use Kerberos to satisfy them.

### A. AUTHENTICATION REQUIREMENTS

In the traditional model, a consumer finds a restaurant, places an order, and then waits for the restaurant to bring the food to the door. With the development of e-commerce, consumers gradually become accustomed to ordering dinners through Apps or websites with great convenience. Soon afterwards, they get unsatisfied with just taking reservations or placing pick-up orders from Internet. The growth in home deliveries makes it real for consumers to dine at home with the same quality food from a fine restaurant.

A typical food ordering and home delivery model is the aggregator model, which emerged about twenty years ago [16]. The aggregator platform simply provides menus online, takes orders from customers and routes them to restaurants that handle the delivery themselves.

Authentication requirements of the food ordering and home delivery model include:

- If a customer wants to order a meal from the online platform, the platform needs to verify the customer's identity.
- Since the transaction often occurs between the restaurant and the customer, the restaurant needs to verify the customer's identity.
- The platform and the restaurant need to verify each other. They usually perform authentications in advance online or offline.
- In the aggregator model, when a deliveryman arrives the customer's home, the customer needs to know whether the genuine restaurant assigns this deliveryman to do the delivery.
- The customer needs to verify the ordering platform too. It is dangerous to visit a fake platform.

Mutual authentications in the food ordering and delivery system are always the basic needs. However, complicated procedures are not welcome. Suppose the customer want to order a meal. If the customer is asked to sign in once to access the platform, that is okay. But if the customer is then asked to sign in to the restaurant's portal again, the customer will feel unhappy. Plus if every restaurants ask the customer to sign in again and again, the system will be disappointing. Therefore, the system should avoid unnecessary repeated authentication.

### B. KERBEROS PROTOCOL

Next, we shall technically review the mechanism of the Kerberos protocol.

Kerberos is a network authentication protocol, which provides mutual authentication for the client and the server over a non-secure network. The Kerberos protocol assumes that there is a trusted third party [6], which is named as a Kerberos server. The Kerberos server consists of an Authentication Server (AS) and a Ticket-Granting Server (TGS). The *AS* and *TGS* are respectively responsible for creating and issuing tickets to the clients upon request. They usually run

on a same physical server named as the Key Distribution Center (KDC). Clients and servers are called principals. Server principals are composed of a primary name, instance, and realm, while client principals do not have to have an instance.

We list a set of abbreviations in Table 1 to help eliminate confusion in the rest of paper.

**TABLE 1. Abbreviations used in the Kerberos protocol.**

| Abbreviations | Descriptions |
|---|---|
| $KDC$ | Key distribution center |
| $AS, as$ | Authentication server |
| $TGS, tgs$ | Ticket-granting server |
| $SS, ss$ | Service server |
| $TGT, tgt$ | Ticket-granting ticket |
| $SGT, sgt$ | Service-granting ticket |
| $C, c$ | Client |
| $ID_x$ | $x$'s $ID$ |
| $ADDR_x$ | $x$'s network address |
| $LIFE$ | Lifetime |
| $STAMP$ | Timestamp |
| $K_x$ | $x$'s private key |
| $K_{x,y}$ | Session key for $x$ and $y$ |
| $\{abc\}K_x$ | "$abc$" encrypted with $x$' key |
| $T_{x,y}$ | $x$'s ticket to $y$ |
| $A_x$ | Authenticator of $x$ |
| $x\_REQ$ | Request from client to $x$ |
| $x\_RESP$ | Response from $x$ to client |

Fig.1 describes the complete Kerberos authentication protocol. There are mainly six interactive messages needed during the authentication as follows:

① $AS\_REQ : ID_c, ID_{tgs}, ADDR_c, LIFE$
② $AS\_RESP : \{K_{c,tgs}, ID_{tgs}, STAMP, LIFE\}K_c$, $\{T_{c,tgs}\}K_{tgs}$
③ $TGS\_REQ : ID_{ss}, \{A_c\}K_{c,tgs}, \{T_{c,tgs}\}K_{tgs}$
④ $TGS\_RESP : \{K_{c,ss}, ID_{ss}, STAMP, LIFE\}$ $K_{c,tgs}, \{T_{c,ss}\}K_{ss}$
⑤ $SS\_REQ : \{A_c\}K_{c,ss}, \{T_{c,ss}\}K_{ss}$
⑥ $SS\_RESP : \{A_{ss}\}K_{c,ss}$

where $T_{c,x}$ includes $ID_c$,$ID_x$,$ADDR_c$,$STAMP$,$LIFE$, and $K_{c,x}$. $A_x$ includes $ID_x$ and $STAMP$.

The Kerberos authentication process consists of three phases (see Fig.1).

1) In the first phase, the client makes a request (message ①) for a $TGT$ from the $AS$. The $AS$ responds with a $TGT$ ($T_{c,tgs}$), and an encrypted session key needed for the next phase. The session key can be decrypted only by a client that possesses the user's password, which has never been communicated over the network. The first phase is used only when the user first logs in to the system.
2) In the second phase, the client presents the $TGT$ to the $TGS$ (message ③), which responds with a ticket $SGT$ ($T_{c,ss}$) and a second session key encrypted with the first session key (message ④). The second phase is indispensable whenever a user authenticates to a new verifier.
3) In the final phase, the $SGT$ as a credential is presented to the given server (message ⑤), which then grants access to the service. Message ⑥ is optional and used only when the client requires mutual authentication.

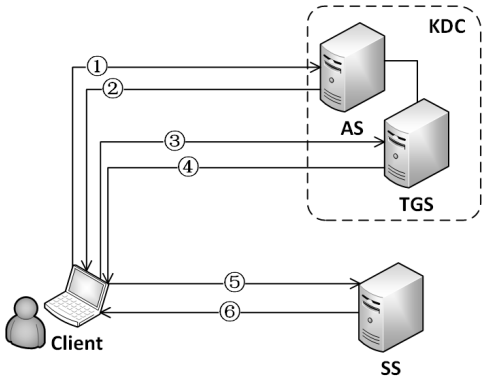


**FIGURE 1. Simplified communication diagram of the Kerberos protocol.**

The session key shared by the client and server is used to authenticate the client and may optionally be used to authenticate the server. It can also be used to encrypt further communication between the two principals or to exchange a separate sub-session key to be used to encrypt further communication.

The Kerberos server maintains a database of principals and their secret keys. The authentication exchanges mentioned previously require read-only access to the Kerberos database. Occasionally, the entries in the database need to be modified, such as when adding new principals or updating a principal's key.

The Kerberos protocol enjoys considerable advantages in network security including confidentiality, integrity, and protect against eavesdropping or replay attacks.

The ticket is sent over a non-secure network and might be intercepted and reused by an attacker. Therefore, an authenticator encrypted with the session key is sent to prove that the message originates from the principal. The authenticator also includes a timestamp, which proves that the message was recently generated and is not a replay. No one except the requesting principal and the server knows the session key. So encrypting the authenticator with the session key proves that it was generated by a principal possessing this session key.

The integrity of the messages exchanged between principals can also be guaranteed by using the session key. It is accomplished by generating and transmitting a message authentication code (MAC) which is a digest function of the client's message keyed with this session key. Confidentiality and privacy can be secured by encrypting data using the session key or the sub-session key packaged in the authenticator.

Moreover, Kerberos has three additional advantages:

- SSO
  Single Sign-On (SSO) is an important feature of Kerberos. With SSO, the client only needs to type password only once. A credential cache on a client machine stores tickets obtained by a client, such as the *TGT* s and *SGT* s. So when the client wants to access to an old service, only phases 3) are replayed. And if the client wants to access to a new service, only phases 2) and 3) are replayed.
- Little assumption
  A trusted third party authentication service is needed. The credential cache must be secured to prevent impersonation. Except these assumptions, Kerberos does not base trust on the host address.
- Anonymity support
  Anonymous user authentication is a common task [17], [18]. Kerberos uses an option to allow asking the TGS not to include the client's name when constructing a ticket but the generic string "USER" [19]. Thus, it offers anonymity support if needed.
- Extensibility
  In the Kerberos environment, the application is free to choose whatever protection may be necessary. For instance, though Kerberos usually builds on the symmetric key cryptography, it supports authentication for users registered with public key certifications. As another instance, many applications use Kerberos upon the initiation of a network connection. Once authenticated, principals can use credentials to start another network or application protocol.

When comparing the security requirements of the aggregator model mentioned previously and the network security provided by the Kerberos protocol, one would find a close similarity. Next, we shall show how to apply Kerberos to the aggregator model.

### C. SCHEME FOR THE AGGREGATOR MODEL

Before we describe the authentication process in detail, we give out some abbreviations in Table 2 which will be used soon.

**TABLE 2. Abbreviations in the ordering and home delivery platform.**

| Entity name | Abbreviation |
|---|---|
| Ordering platform | $OP$ or $op$ |
| Home delivery platform | $DP$ or $dp$ |
| Customer | $CU$ or $cu$ |
| Supplier | $SU$ or $su$ |
| Deliveryman | $D$ or $d$ |
| Phone number | $PN$ |

To distinguish the abbreviations of "client" and "customer", we use "*cu*" as the abbreviation of "customer". Similarly, we use "*su*" as the abbreviation of "supplier". Suppliers may be flower shops, restaurants, or book retail stores. Customers could order flowers, fast foods or books from them.

We compare the security requirements of the aggregator model and the network security requirement of the client/server model in the Kerberos environment, see Table 3. Despite the offline feature, the aggregator model shows close similarities to the client/server model. It is suitable for the aggregator model to use Kerberos as its authentication solution. Kerberos is such a widely deployed network protocol that we can design the authentication scheme for the aggregator model without much difficulty.

**TABLE 3. Requirement comparison between the aggregator model and the client/server model.**

| | Aggregator model | C/S model |
|---|---|---|
| Mutual authentication | needed | needed |
| Confidentiality | needed | needed |
| Integrity | optional | needed |
| Single Sign-On | needed | needed |
| Trusted third party | needed | needed |

We carefully listed the Kerberos principals and their devices, along with their aggregator counterparts. As shown in Table 4, the aggregator model should be a scheme based on the trusted third-party. And the platform which is trusted by customers and suppliers is the best candidate for the role.

**TABLE 4. Principal and device comparison between the aggregator model and the client/server model.**

| | Aggregator model | C/S model |
|---|---|---|
| Trusted third party | platform | KDC |
| Service applier | customer | client |
| Service provider | restaurant/deliveryman | server |
| Device of applier | smartphone | computer |
| Device of supplier | smart phone/computer | computer |

Likewise, we describe the authentication process in three phases.

In the first phase as shown in Fig.2(a), the customer makes a request (message ①) for a *TGT* from the platform. There is a growing trend that more and more customers use smart phones instead of computers to makes the request. The platform responds with a *TGT* (message ②), and an encrypted session key needed for the next phase. The session key can be decrypted only by the customer who possesses the customer's password, which has never been communicated over the network. The first phase is used only when the customer first signs in to the system.

In the second phase shown in Fig.2(b), the customer presents the *TGT* to the *TGS* (message ③), which responds with a service-granting ticket *SGT* and a second session key encrypted with the first session key (message ④). The second phase is indispensable whenever a customer authenticates to a new supplier.

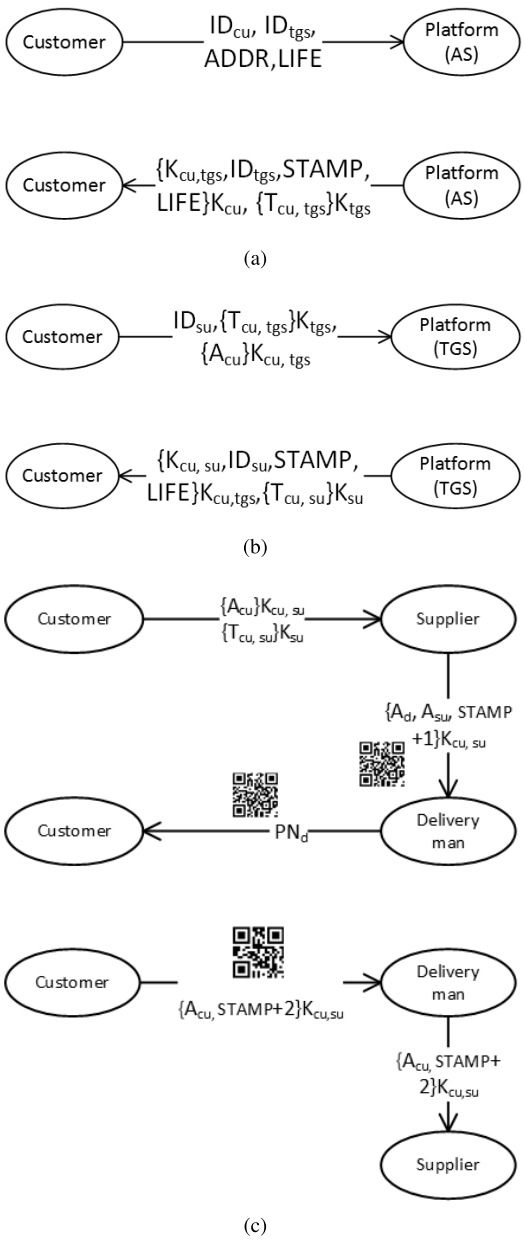


**FIGURE 2.** The first/second/third phase of the Kerberos based authentication scheme.

In a networking context, authentication is the act of proving identity to a network application or resource. The situation does not change in the communication among the customer's smart phone, the platform server, and the supplier's computer.

In the final phase shown in Fig.2(c), the *SGT* as a credential is presented to the given supplier (message ⑤). The supplier verifies the signed data with the session key to validate the authentication attempt.

Unlike typical online applications of Kerberos, the deliveryman, and the customer need offline face-to-face authentication in the delivery business. We compared the implementation between the electronic and face-to-face authentication in Table 5. In the real world, one prefers

**TABLE 5.** Comparison between electronic and face-to-face authentication.

| | Electronic | Face-to-face |
|---|---|---|
| Authentication type | online | offline |
| Message carrier | data stream | bar-code image |
| Message size | unlimited | limited |
| Encryption/Decryption | needed | needed |
| Principal's address | IP address | phone number |

scanning a bar-code image rather than starting up a computer to send or receive a message. A recommended practice is sticking some bar-code on the dinnerware. The authenticator $A_d$ in the bar-code includes $PN_d$ and *STAMP*, where $PN_d$ is the phone number of the deliveryman.

Theoretically, a bar-code image contains data of any finite size with enough high resolution. However, the recognizability of the smart phone is limited. Fig.3 shows that the bar-code image of 2048 bits data has a much higher resolution than the bar-code image of 24 bits. Therefore, very large size of message is not preferable. We use less than 384 bits in the real project.


**FIGURE 3.** Two 2D bar-code images with different data sizes.

The bar-code can be replaced with the RFID which is harder to replicate and can provide larger storage. One can integrate the face-to-face authentication scheme into the door access and guard system without much difficulty. Thus, the physical security can be significantly enhanced.

### D. SECURITY

Since Butler *et al.* [8] have formally verified the design of Kerberos, we design the Kerberos based scheme as rigorously as possible to reap the full benefits of security of Kerberos authentication. These schemes allow nodes communicating over the non-secure environment to prove their identity to one another in a secure manner. Additionally, the face-to-face authentications are implemented through bar-codes authentication, which is a convenient way in the real world. All the data in the bar-codes are encrypted by the principals' secret keys or session keys. Therefore, we need not worry about some challenges such as eavesdropping and replay attacks.

For Man-At-The-End attacks [20], if all principals keep secret keys secret and nobody is threatened with physical violence, the scheme is secure. If an attacker generates a bar-code copy, just like eavesdrops on the network, the attacker will be found by checking the phone number or other

identification information in the ticket. The security can be significantly enhanced if we enable further authentications based on the session key shared between, or use additional biometric authentications.

### E. LIMITATION

Our schemes inherit the limitation of the Kerberos as well as its advantage.

The limitations mainly include:

- Keeping secret keys secret
  Principals must keep their secret keys secret. If an intruder steals a principal key, the intruder will be able to masquerade as that principal or impersonate any server to the legitimate principal. For the same reason, our schemes are vulnerable to password guessing attacks.
- Vulnerability to DOS
  Kerberos has not solved the Denial Of Service (DOS) attacks. Therefore, the online part of our scheme is vulnerable to DOS.
- Time synchronization
  Each principal on the network must have a clock, which is loosely synchronized to the time of the other principals.

We make these assumptions on the environment in which our scheme can properly function.

### F. AUXILIARY PROTOCOLS AND OPTIONAL BEHAVIOR

The recipient may check whether the sender's address (or cell phone number) matches the supposed sender's address in the message, and whether the recipient's addresses (or cell phone number) matches the supposed recipient's address in the message [6]. In fact, the address check does not provide any added security, since the identity has already been verified before. Thus, from the perspective of security, the recipient may ignore the address even if it is presented. Nevertheless, the address should be included in the authenticators as business information, or for audit.

With the mutual secret key in hand, principals in the aggregator model can also exchange any message encrypted with their shared key.

## III. GENERALIZED AUTHENTICATION PATH AND CROSS-REALM AUTHENTICATION SCHEMES

With the rise of crowdsourcing economies, the crowdsourcing delivery recently emerges as the mainstream implementation of home delivery. Advantages of using crowdsourcing may include decreased costs and increased flexibility [21]. In China, even lower-end traditional delivery restaurants have found it more cost efficient to crowdsource distribution logistics [22].

The Kerberos protocol is also designed for operating across realms [6]. In this section, we will technically review the mechanism of the cross-realm Kerberos. Then we propose the generalized authentication path, and design an authentication scheme for the crowdsourcing delivery model.

### A. CROSS-REALM KERBEROS

We describe the cross-realm authentication in a typical scenario. Suppose a user in realm 1 wants to print a file, but there is no printer in realm 1. Fortunately, the *KDC* server in realm 1 has full accessibility of realm 2. But there is no printer in realm 2 too. Next, the *KDC* server of realm 2 has full accessibility of realm 3, and the *KDC* server of realm 3 has full accessibility of realm 4, and so on. Finally, The *KDC* server of realm $N-1$ has full accessibility of realm $N$, and there is a printing service with a printer in realm $N$.

We illustrate the authentication process in Fig.4 for this scenario. The interactive messages needed during the authentication are presented as follows:

1) Exchange with $AS1/TGS1$ of realm 1
   $AS1\_REQ : ID_{c@r1}, ID_{tgs1}, ADDR_{c@r1}, LIFE$
   $AS1\_RESP : \{K_{c@r1,tgs1}, ID_{tgs1}, STAMP, LIFE\}$
   $K_{c@r1}, \{T_{c@r1,tgs1}\}K_{tgs1}$
   $TGS1\_REQ : ID_{as2}, \{A_{c@r1}\}K_{c@r1,tgs1},$
   $\{T_{c@r1,tgs1}\}K_{tgs1}$
   $TGS1\_RESP : \{K_{c@r1,as2}, ID_{as2}, STAMP, LIFE\}$
   $K_{c@r1,tgs1}, \{T_{c@r1,as2}\}K_{as2}$
2) Exchange with $AS2/TGS2$ of realm 2
   $AS2\_REQ : ID_{tgs2}, \{A_{c@r1}\}K_{c@r1,as2},$
   $\{T_{c@r1,as2}\}K_{as2}$
   $AS2\_RESP : \{K_{c@r1,tgs2}, ID_{tgs2}, STAMP, LIFE\}$
   $K_{c@r1,as2}, \{T_{c@r1,tgs2}\}K_{tgs2}$
   $TGS2\_REQ : ID_{as3}, \{A_{c@r1}\}K_{c@r1,tgs2},$
   $\{T_{c@r1,tgs2}\}K_{tgs2}$
   $TGS2\_RESP : \{K_{c@r1,as3}, ID_{as3}, STAMP, LIFE\}$
   $K_{c@r1,tgs2}, \{T_{c@r1,as3}\}K_{as3}$
   .....
3) Exchange with $ASn/TGSn$ of realm n
   $ASn\_REQ : ID_{tgsN}, \{A_{c@r1}\}K_{c@r1,asN},$
   $\{T_{c@r1,asN}\}K_{asN}$
   $ASn\_RESP : \{K_{c@r1,tgsN}, ID_{tgsN}, STAMP, LIFE\}$
   $K_{c@r1,asN}, \{T_{c@r1,tgsN}\}K_{tgsN}$
   $TGSn\_REQ : ID_{p@rN}, \{A_{c@r1}\}K_{c@r1,tgsN},$

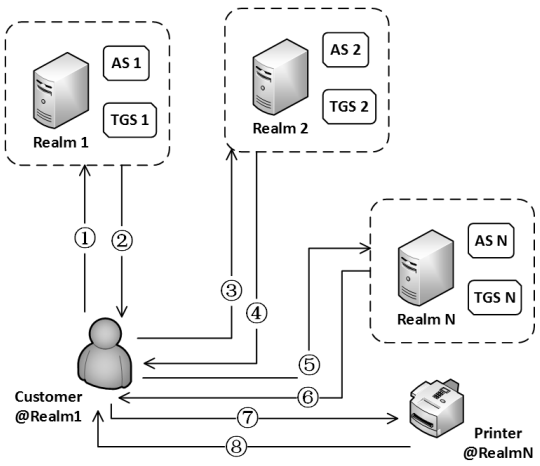


**FIGURE 4.** Simplified communication diagram of the cross-realm Kerberos protocol.

$\{T_{c@r1,tgsN}\}K_{tgsN}$
$TGSn\_RESP : \{K_{c@r1,p@rN}, ID_{p@rN}, STAMP, LIFE\}K_{c@r1,tgsN}, \{T_{c@r1,p@rN}\}K_{p@rN}$

4) Request to the printing service
$P@Rn\_REQ : \{A_{c@r1}\}K_{c@r1,p@rN}, \{T_{c@r1,p@rN}\}K_{p@rN}$
5) Response from the printing service
$P@Rn\_RESP : \{A_{p@rN}\}K_{c@r1,p@rN}$

where $c@r1$ and $p@rN$ are the abbreviations of $customer@realm1$ and $printer@realmN$ respectively. We omit the authentication process from realm 3 to realm $N-1$ to avoid unnecessary repetition.

A cross-realm authentication starts when the client sends a request $AS1\_REQ$ to its home $AS1$ in order to obtain a ticket $TGT1$. This ticket is used in a follow-up message exchange with the home $TGS1$ where the client requests a special ticket called **cross-realm ticket** $\{T_{c@r1,as2}\}K_{as2}$. The cross-realm ticket, which is protected with the inter-*KDC* key ($K_{c@r1,as2}$) shared between realm 1 and 2, now is used to communicate with the $AS2$ in realm 2. When realm 2 shares an inter-realm key with the next realm, $AS2$ will issue a ticket $\{T_{c@r1,tgs2}\}K_{tgs2}$ that can be used to acquire a ticket for visiting next realm. Again and again, finally the client obtains a ticket $\{T_{c@r1,p@rN}\}K_{p@rN}$ for accessing the printing service in the last realm. The printing service decrypts the ticket using its private key $K_{p@rN}$, extracts the $K_{c@r1,p@rN}$ from the ticket, decrypts the client's authenticator $A_{p@rN}$ and checks it. If the authenticator passes, then the client will access the printing service finally.

The end-service often wants to know which realms were transited in the authentication process. To facilitate this decision, a field in each ticket optionally contains the names of the realms involved.

### B. AUTHENTICATION PATH AND ITS GENERALIZATION

The cross-realm Kerberos have been successfully applied to many network security scenarios. However, it cannot be applied to the crowdsourcing platform directly, because there is essential difference between these two scenarios.

If the customer requires all cross-realm authentications, we can apply the typical cross-realm Kerberos scheme without much change. Working on this assumption, the customer orders a meal from the ordering platform. Next, the customer finds a deliveryman from a delivery platform introduced by the restaurant. Then the deliveryman goes to the restaurant to take the meal. However, in the real world, the customer would not like to do so many things. What the customer want to do is just ordering a meal online and paying for the food fee and the delivery fee. The customer does not care how to arrange the delivery. Therefore, we have to modify the typical cross-realm authentication scheme.

*Definition 1 (Authentication Path [6]):* In the Kerberos, a sequence of intermediate realms transited in the authentication process when communicating from one realm to another has been defined as the authentication path.

We can easily obtain the authentication path even for a hierarchically-configured Kerberos [6]. The authentication path between the client and the print server have been traversed in Fig.4 is ($Realm1, Realm2, \cdots, RealmN$), or

$$(R_1, R_2, \cdots, R_N)$$

for short.

We illustrate how to construct the authentication path in Fig.5. The client, KDC, and the server trust mutually in each realm. We simplify their links, keep them still connected with KDC in center, as shown in Fig.5(a). Then the cross-realm printing service in Fig.4 can be modeled as a linked list in Fig.5(b).

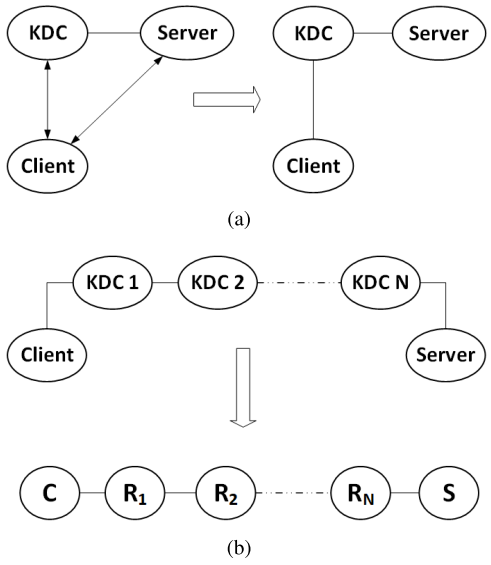


**FIGURE 5.** Constructing the authentication path for the cross-realm printing service.

So far, the authentication path in published literature is always a linked list from client to server, without exception. However, we believe that some modification may be beneficial and applicable.

*Definition 2 (Generalized Authentication Path):* The generalized authentication path is defined as a permutation of the realms sequence from client to server.

**Algorithm 1** Constructing the Generalized Authentication Path

**Require:** the linked list $P_{in}$ from client to server.
Set the generalized authentication path $P_{out}$ to empty.
Let $P = P_{in}$.
**while** $P$ is not empty **do**
  Find a realm $R_k$ which plays a role of KDC for its two neighbors.
  Append $R_k$ to $P_{out}$
  Delete $R_k$ from $P$
**end while**
**return** the generalized authentication path $P_{out}$.

The generalized authentication path can be constructed recursively. We describe how to construct the generalized authentication path in Algorithm 1. In each step of the while loop, we can always find a realm which plays a role of KDC for its two neighbors, and one of its neighbors initiate this authentication request (see Fig.6).

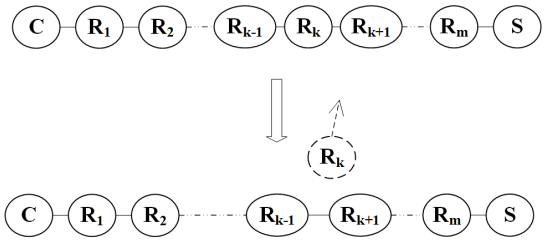


**FIGURE 6.** Each step in the recursive construction of generalized authentication path.

### C. PROVABLE SECURITY

Next, we present the statements that provide the basic security guarantees for the generalized cross-realm Kerberos we proposed previously.

1) Confidentiality of $K_{c@rN,tgsN}$
In a realm $n$ ($n \in \{1, 2, \cdots, m\}$), if the intruder does not know the private keys ($K_{c@rN}$ and $K_{tgsN}$) used to encrypt the session key $K_{c@rN,tgsN}$ generated by the authentication server $ASn$ for use by $C@Rn$ and $TGSn$, then the intruder cannot learn $K_{c@rN,tgsN}$.
2) Confidentiality of $K_{c@rN,s@rN}$
In a realm $n$ ($n \in \{1, 2, \cdots, m\}$), if the intruder knows neither the private key $K_{s@rN}$ used by a $TGSn$ to encrypt the $T_{c@rN,s@rN}$ containing a new session key $K_{c@rN,s@rN}$ for a client to use with a server, nor the session key used by the client to request the $T_{c@rN,s@rN}$, then the intruder cannot learn $K_{c@rN,s@rN}$.
3) Confidentiality of cross-realm $K_{c@rN}$
For a realm $N$ ($N \in \{2, 3, \cdots, M\}$), if the intruder does not know any of the keys $\{K_{c@rI}, K_{tgsI}, K_{s@rI}, K_{c@rI,tgsI}, K_{c@rI,s@rI}\}, I \in \{1, 2, \cdots, N-1\}$ in the realms established previously, then the intruder cannot learn $K_{c@rN}$.
4) Authentication of $ASn$ to client $C@Rn$
In a realm $n$ ($n \in \{1, 2, \cdots, m\}$), if the client $C@Rn$ sees what appears to be a valid reply from the $ASn$ and if the key $K_{c@rN}$ is secret, then the $ASn$ generated this reply to a request that named the client.
5) Authentication of request for $TGTn$
In a realm $n$ ($n \in \{2, 3, \cdots, m\}$), if the $ASn$ processes a request for a ticket $TGTn$ and if neither the key $K_{asN}$ encrypting the ticket nor the key $K_{c@rN}$ shared between $C@Rn$ and $ASn$ is known to the intruder, then the request was generated by the client $C@Rn$. Here the key $K_{asN}$ or $K_{c@rN}$ may be generated from previous realms.
6) Authentication of request for $T_{c@rN,s@rN}$
In a realm $n$ ($n \in \{1, 2, \cdots, m\}$), if the $TGSn$ processes a request for a service granting ticket and if neither the private key $K_{tgsN}$ encrypting the ticket, nor the session key $K_{c@rN,tgsN}$ encrypting the authenticator, nor the private key $K_{c@rN}$ which encrypted the session key $K_{c@rN,tgsN}$ is known to the intruder, then the ticket in the request is a valid ticket and was generated by the $ASn$ with whom the $TGSn$ shares a private key $K_{tgsN}$. Furthermore, the authenticator included in the request was generated by the client $C@Rn$ named in the ticket.
7) Authentication of request to service server
In a realm $n$ ($n \in \{1, 2, \cdots, m\}$), if the intruder does not know the private key $s@rN$ used to encrypt a $T_{c@rN,s@rN}$ for a client $C@Rn$ to present to a server $S@Rn$, then if $S@Rn$ processes the request, ostensibly from $C@Rn$, containing this $T_{c@rN,s@rN}$ and the encrypted authenticator, then some $TGSn$ generated the session key $K_{c@rN,s@rN}$ for $C@Rn$ to securely communicate with $S@Rn$ and also created the $T_{c@rN,s@rN}$. Furthermore, if the intruder never learns the session key $K_{c@rN,tgsN}$ which the $TGSn$ used to encrypt $K_{c@rN,s@rN}$ when sending the $T_{c@rN,s@rN}$ to $C@Rn$, then $C@Rn$ created the authenticator.
8) Structural soundness
In the final realm $M$, if the service server $S@Rm$ processes a request for service from a client $C@Rm$, and if the keys $K_{c@rI}$ , $K_{tgsI}$, $K_{c@rI,tgsI}$, $K_{c@rI,s@rI}$, $K_{s@rI}$ ($I \in \{1, 2, \cdots, M\}$) are all secret, then exchanges in the previous $M-1$ realms happened and did so in the expected order.

These properties cover the authentication, confidentiality, and structural soundness of our generalized Kerberos. And they are much the same as properties of the classical cross-realm Kerberos. By using the MSR language [23], Butler *et al.* [8] have provided sketch proofs that are also applicable here. These proofs are complete with supplementary theorems and proofs in [23], [24], and [25].

### D. AUTHENTICATION SCHEMES FOR THE CROWDSOURCING DELIVERY MODEL

Following Algorithm1, we can construct the generalized authentication path for the crowdsourcing delivery model. The path is $\{R_1, R_3, R_2\}$ or $\{R_{op}, R_{dp}, R_{su}\}$ as shown in Fig.7.

In this model, the supplier has registered to the ordering platform and the delivery platform online or offline. The client initiate an authentication request to the ordering platform for establishing connection with the supplier securely. If the order is confirmed, the supplier posts a delivery requirement on the delivery platform. Then a deliveryman sends the second authentication request to the delivery platform for this delivery order. If the deliveryman wins the delivery bid, the third authentication request is sent from the deliveryman to the supplier. The new realm establishes a secured connection between the deliveryman and the customer.

There is a major difference between this structure and the nested structure of the typical cross-realm scenario in Fig.4. In this scheme, we generate a new *KDC* from a principal (the supplier) which has been a server of the previous realm.

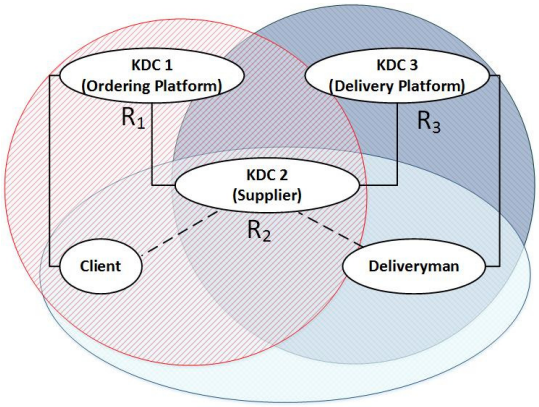


**FIGURE 7.** Realms in the crowdsourcing delivery model.

We provide a detailed description of the three step cross-realm authentication in the crowdsourcing delivery model, as shown in Fig.8.

Fig.8(a) depicts the ordering process. This process is performed on the Internet among the smart phone of the customer, the computer of the supplier, and the server of the online ordering platform. Messages in the authentication process include:

① $AS1\_REQ : ID_{cu}, ID_{tgs1}, ADDR, LIFE$
② $AS1\_RESP : \{K_{cu,tgs1}, ID_{tgs1}, STAMP, LIFE\}$
$K_{cu}, \{T_{cu,tgs1}\}K_{tgs1}$
③ $TGS1\_REQ : ID_{su}, \{A_cu\}K_{cu,tgs1},$
$\{T_{cu,tgs1}\}K_{tgs1}$
④ $TGS1\_RESP : \{K_{cu,su}, ID_{su}, STAMP, LIFE\}$
$K_{cu,tgs1}, \{T_{cu,su}\}K_{su}$
⑤ $SU\_REQ : \{A_{cu}\}K_{cu,su}, \{T_{cu,su}\}K_{su}$

where the *KDC* established by the ordering platform is composed of $AS1$ and $TGS1$. Message ① and message ② are used only when the user first signs in to the system. The ticket $\{T_{cu,tgs1}\}K_{tgs1}$ could be stored in the smart phone for a preset period. Optionally, the customer can also verify the supplier by using shared key after step ⑤.

Fig.8(b) describes how the deliveryman gets the delivery order from the supplier. This process is operated on the Internet among the smart phone of the deliveryman, the computer of the supplier, and the server of the delivery platform. Messages in this process include:

① $AS3\_REQ : ID_d, ID_{tgs3}, ADDR, LIFE$
③ $AS3\_RESP : \{K_{d,tgs3}, ID_{tgs3}, STAMP, LIFE\}$
$K_d, \{T_{d,tgs3}\}K_{tgs3}$
③ $TGS3\_REQ : ID_{su}, \{A_d\}K_{d,tgs3},$
$\{T_{d,tgs3}\}K_{tgs3}$
④ $TGS3\_RESP : \{K_{d,su}, ID_{su}, STAMP, LIFE\}$
$K_{d,tgs3}, \{T_{d,su}\}K_{su}$
⑤ $SU\_REQ : ID_{cu}, \{A_d\}K_{d,su}, \{T_{d,su}\}K_{su}$
⑥ $SU\_RESP : \{A_{su}\}K_{d,su}$

where the *KDC* established by the delivery platform is composed of $AS3$ and $TGS3$. The deliveryman has not arrived in this stage, though he has won the delivery bid.

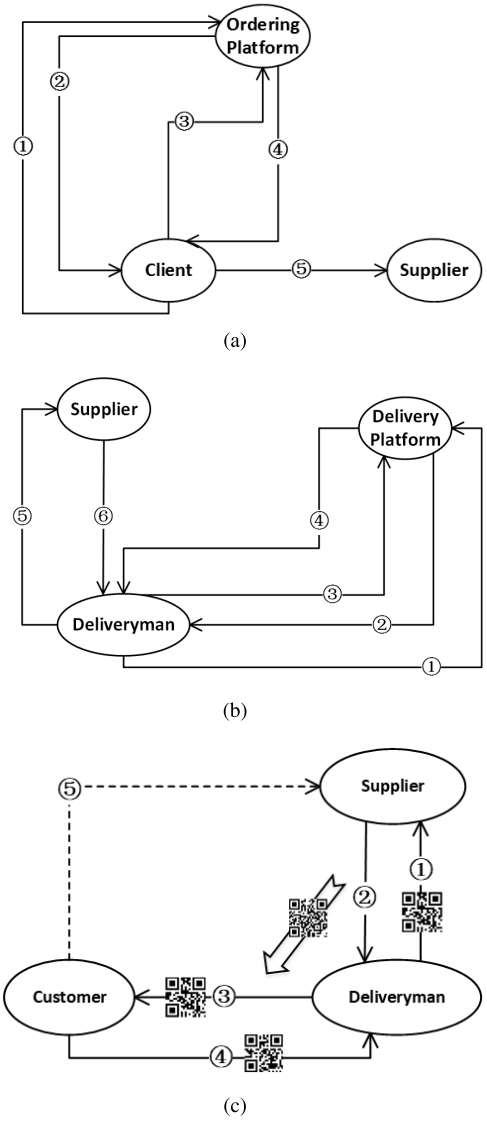


**FIGURE 8.** The cross-realm authentication in the crowdsourcing delivery model.

Fig.8(c) describes the delivery process. The smart phone of the deliveryman, the computer of the supplier, and the smart phone of the customer participate in this process. Messages in this process include:

① $SU\_REQ : \{ID_d, ID_{cu}, ADDR, LIFE\}K_{d,su}$
② $SU\_RESP : \{K_{d,cu}, A_{su}, ID_{cu}, STAMP, LIFE\}$
$K_{d,su}, \{T_{d,cu}\}K_{cu,su}$
③ $CU\_REQ : \{A_d\}K_{d,cu}, \{T_{d,cu}\}K_{cu,su}$
④ $CU\_RESP : \{A_{cu}, STAMP + 1\}K_{d,cu}$

where the *KDC* is established by the supplier. The *KDC* can be implemented as a simplified Kerberos without *TGS*, because *TGT* is used only once in most cases.

The supplier and the deliveryman need a mutually offline authentication. They use bar-code image in smart phone to perform the face-to-face verification. The restaurant (supplier) sticks the bar-code of the encrypted deliveryman-customer ticket on the dinnerware. The customer scans the

bar-code to get the session key with the deliveryman. Then the customer and the deliveryman can perform a mutually authentication base on their sharing session key. Before the stage is over, the customer notifies the supplier that the food has been received, by sending message ecrypted with their shared key.

The authentication processes in above three stages aggregate in Fig.9. As shown in Fig.9, we add a message⑫ which is a message of order confirmation encrypted with $K_{cu,su}$. Through this message, the customer is notified that some deliveryman has accepted the delivery order and is supposed to make the delivery soon. Also, we add a message⑰ encrypted with $K_{cu,su}$ as a confirmation. Through this message, the supplier can be notified that the customer has received the food. Message⑫ and ⑰ are messages for business rather than for security.

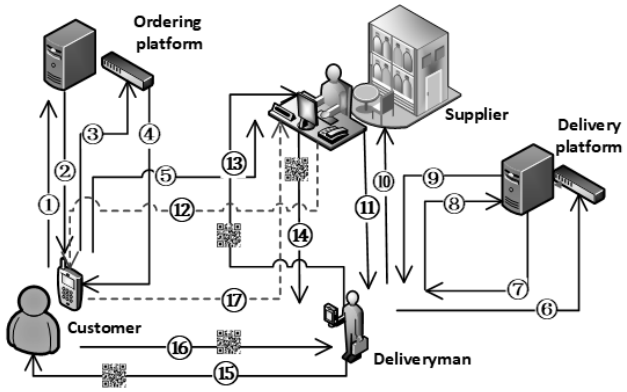


**FIGURE 9.** Communication diagram of the ordering and home delivery business.

Sometimes the ordering platform and the delivery platform are united and operated by a company. That does not matter. Even in this case, the ordering system and the delivery system are usually operated by different departments.

## IV. MORE SOPHISTICATED APPLICATIONS

The order and home delivery business is undergoing continuous changes nowadays. A wide variance appears recently [22]. In this section, we mainly discuss three common variances in the crowdsourcing environment.

### A. RECEIVING AGENT MODEL

A customer may wish to use the services of a receiving agent for privacy. For example, a person running a home-based business may not wish to disclose the residential address. Depending on the agreement between the customer and the receiving agent, the receiving agent can forward the goods to the customer or hold it for pickup. In another case, a deliveryman is not allowed to enter an office building for the sake of security. The deliveryman leaves the goods to the receptionist or the self-service drop box there.

The generalized authentication path of this model is $\{R_{op}, R_{dp}, R_{su}, R_{cu}\}$.

Fig.10 illustrates the additional authentication process of the receiving agent model. When the deliveryman wants to let the agent believe he/she is the genuine deliveryman, the deliveryman exchanges messages with the customer, and obtains a session key with the agent. Subsequent bar-code authentications are performed by using this session key. The deliveryman uses message⑦ to notify the customer that the goods have already been sent to the agency.

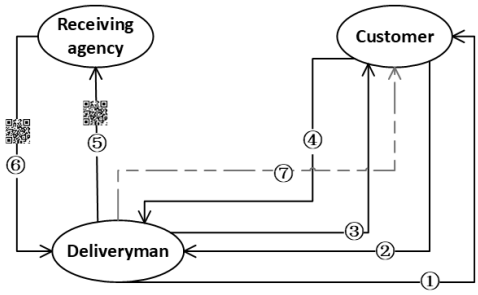


**FIGURE 10.** The added authentication part of the receiving agent model.

### B. UNIFIED SERVICE OF FRANCHISED CHAIN MODEL

A franchise retail chain shares a brand and central management, and usually builds a standard format through architectural prototype development and offers a standard menu or services. In many service categories such as restaurant chain, chain businesses have come to dominate the market in many parts of the world.

The generalized authentication path of this model is $\{R_{op}, R_{us}, R_{dp}, R_{su}\}$, where "$us$" is the abbreviation of "unified supplier".

Fig.11 illustrates the additional authentication process of the franchised chains model. The franchised chains provide service through the franchiser's unified portal. Mutual trust between the franchiser and the franchisees has been established in advance. When the customer places an order, the franchiser posts the order on the portal or through the inner channel. A franchisee nearby accepts the order and becomes the end supplier.

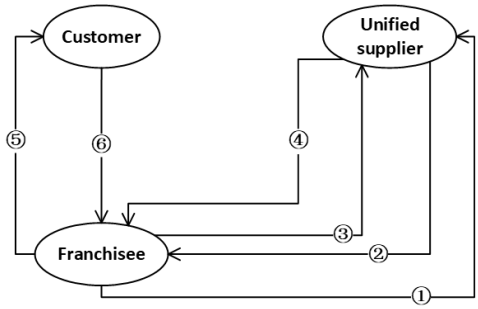


**FIGURE 11.** The added authentication part of the franchised chains model.

### C. RELAY DELIVERY MODEL

A relay delivery collects payment for the order. However, as the relay delivery service cannot fulfill the delivery order itself unless the delivery is local to the location. It relays the order and payment to a local delivery company or a local deliveryman in the delivery area, minus a commission.

The generalized authentication path of this model is $\{R_{op}, R_{dp}, R_{d1}, R_{dp}, R_{d2}, \cdots, R_{dp}, R_{dn}, R_{su}\}$.

Fig.12 illustrates the additional authentication process of the relay delivery model. The latter deliveryman exchanges messages with the delivery platform to obtain a sharing session key with the former deliveryman. Then the latter deliveryman, instead of the former deliveryman, is supposed to execute the delivery order. Finally, the last deliveryman accepts the delivery. And then bar-code authentications are performed between the last deliveryman and the customer. Message⑦,⑯, and ⑲ in Fig.12 are notifying messages.

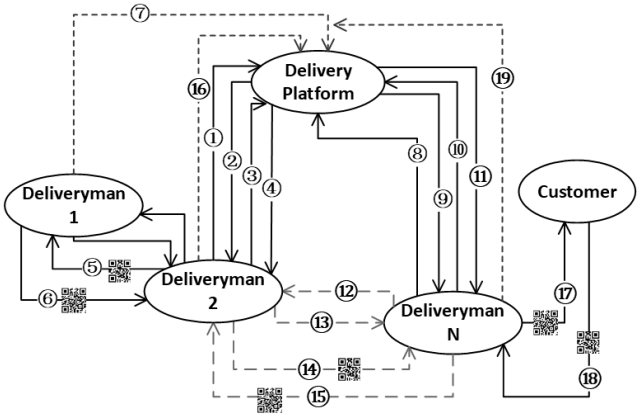


**FIGURE 12.** The added authentication part of the relay delivery model.

Besides the above three models in this section, Other predictable models such as *virtual restaurant* and *group customers* can find authentication schemes likewise.

## V. CONCLUSION

We generalized the authentication path of the Kerberos protocol, which is a technically mature and structurally sound network protocol. The generalization makes it possible to design cross-realm authentication scheme for many types of complicated business.

We integrated the Kerberos into the online ordering and home delivery platform. The desired goal is to change the present situation that strict authentication is ignored in the real world. We combined Kerberos authentication and bar-code authentication in the schemes of the aggregator model and the crowdsourcing model. Moreover, we extended the schemes to three more complicated business models: the receiving agent model, the franchised chains model, and the relay delivery model. Emphasis was placed on the detailed schemes and the establishment of the generalized authentication path.

We are making effort in designing and justifying authentication solutions for more concrete business models. Future work also involves attempts on applying other network protocols to the offline real-life business models.

## REFERENCES

[1] Statista Ltd., (2017). *Food delivery industry in the U.S.—Statistics & Facts*. [Online]. Available: https://www.statista.com/topics/1986/food-delivery-industry-in-the-us/

[2] Abc. Ltd., (2016). *Houston Homeowner Brutally Beaten by Fake UPS Delivery Driver*. [Online]. Available: http://abc7.com/news/texas-homeowner-brutally-beaten-by-fake-ups-delivery-driver/1637671/

[3] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 33–38, Sep. 1994.

[4] S. P. Miller, B. C. Neuman, J. I. Schiller, and J. H. Saltzer, "Kerberos authentication and authorization system," Project Athena Tech. Plan Sect. E.2.1, MIT Project Athena, Cambridge, MA, USA, Dec. 1987, pp. 1–36. [Online]. Available: http://web.mit.edu/Saltzer/www/publications/athenaplan/e.2.1.pdf

[5] J. Kohl and B. Neuman, *The Kerberos Network Authentication Service (v5)*, document RFC 1510, 1993. [Online]. Available: https://tools.ietf.org/html/rfc1510.html

[6] J. Kohl and B. Neuman, *The Kerberos Network Authentication Service (v5)*, document RFC 4120, 2015. [Online]. Available: https://tools.ietf.org/html/rfc4120.html

[7] N. Jrgensen, "Incremental and decentralized integration in FreeBSD," in *Perspectives on Free and Open Source Software*, Cambridge, MA, USA: MIT Press, 2005, pp. 227–244.

[8] F. Butler, I. Cervesato, A. D. Jaggard, A. Scedrov, and C. Walstad, "Formal analysis of Kerberos 5," *Theor. Comput. Sci.*, vol. 367, nos. 1–2, pp. 57–87, 2006.

[9] I. Cervesato, A. D. Jaggard, A. Scedrov, J.-K. Tsay, and C. Walstad, "Breaking and fixing public-key Kerberos," *Inf. Comput.*, vol. 206, nos. 2–4, pp. 402–424, 2008.

[10] P. D. Rowe, J. D. Guttman, and M. D. Liskov, "Measuring protocol strength with security goals," *Int. J. Inf. Secur.*, vol. 15, no. 6, pp. 575–596, 2016.

[11] H. Liu, P. Luo, and D. Wang, "A distributed expansible authentication model based on Kerberos," *J. Netw. Comput. Appl.*, vol. 31, no. 4, pp. 472–486, 2008.

[12] J. Lopez, R. Oppliger, and G. Pernul, "Authentication and authorization infrastructures (AAIs): A comparative survey," *Comput. Secur.*, vol. 23, no. 7, pp. 578–590, 2004.

[13] G. Lawton, "Open source security: opportunity or oxymoron?" *Computer*, vol. 35, no. 3, pp. 18–21, Mar. 2002.

[14] F. Pereñíguez-García, R. Marín-López, G. Kambourakis, A. Ruiz-Martínez, S. Gritzalis, and A. F. Skarmeta-Gómez, "KAMU: Providing advanced user privacy in Kerberos multi-domain scenarios," *Int. J. Inf. Secur.*, vol. 12, no. 6, pp. 505–525, 2013.

[15] A. P. Shrestha, D. Y. Choi, G. R. Kwon, and S.-J. Han, "Kerberos based authentication for inter-domain roaming in wireless heterogeneous network," *Comput. Math. Appl.*, vol. 60, no. 2, pp. 245–255, 2010.

[16] A. K. Dutt and P. Skott, "Keynesian theory and the aggregate-supply/aggregate-demand framework: A defense," *Eastern Econ. J.*, vol. 22, no. 3, pp. 313–331, 1996.

[17] S. Kumari and M. K. Khan, "More secure smart card-based remote user password authentication scheme with user anonymity," *Secur. Commun. Netw.*, vol. 7, no. 11, pp. 2039–2053, 2015.

[18] F. Wen, W. Susilo, and G. Yang, "A secure and effective anonymous user authentication scheme for roaming service in global mobility networks," *Wireless Personal Commun.*, vol. 73, no. 3, pp. 993–1004, 2013.

[19] L. Zhu, P. Leach, and S. Hartman, *Anonymity Support for Kerberos*, document RFC 6112, 2014. [Online]. Available: https://tools.ietf.org/html/rfc6112.html

[20] A. Akhunzada *et al.*, "Man-at-the-end attacks: Analysis, taxonomy, human aspects, motivation and future directions," *J. Netw. Comput. Appl.*, vol. 48, pp. 44–57, Feb. 2015.

[21] R. Buettner, "A systematic literature review of crowdsourcing research from a human resource management perspective," in *Proc. Hawaii Int. Conf. Syst. Sci.*, Jan. 2015, pp. 4609–4618.

[22] W.-M. To and L. S. L. Lai, "Crowdsourcing in China: Opportunities and concerns," *IT Prof.*, vol. 17, no. 3, pp. 53–59, May/Jun. 2015.

[23] N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov, "Multiset rewriting and the complexity of bounded security protocols," *J. Comput. Secur.*, vol. 12, no. 2, pp. 247–311, 2004.
[24] F. Butler, I. Cervesato, A. D. Jaggard, and A. Scedrov, "A formal analysis of ome properties of Kerberos 5 using MSR," in *Proc. IEEE Workshop Comput. Secur. Found.*, Jun. 2002, p. 165.
[25] I. Cervesato, A. D. Jaggard, A. Scedrov, and C. Walstad, "Specifying Kerberos 5 cross-realm authentication," in *Proc. Workshop Issues Theory Secur.*, Long Beach, CA, USA, 2005, pp. 12–26.

**HUI LI** received the B.S. degree from Xi'an Jiaotong University in 1994, and the M.S. and Ph.D. degrees from Shanghai Jiaotong University in 1997 and 1999, respectively. He was a Visiting Scholar with the University of California at Davis, Davis, CA, USA, in 2013.

He is currently an Associate Professor with the Department of Computer Science and Technology. His research interests include number theory, algebra, and complexity and artificial intelligence.

**YI NIU** received the B.S. degree from Beijing Normal University in 1999. She joined the Digital Development Center, China National Publications Import and Export (Group) Corporation (CNPIEC), in 1999. Since 2012, she has been responsible for the construction of the digital platform and digital resources importing and exporting business.

She is currently the Managing Director of the Digital Development Center of CNPIEC.

**JUNKAI YI** received the B.S. and Ph.D. degrees from the Beijing Institute of Technology in 1993 and 1998, respectively.

He is currently a Professor with the Department of Computer Science and Technology, Beijing University of Chemical Technology.

**HONGYU LI** received the B.S. degree from the Beijing University of Chemical Technology in 2015, where he is currently pursuing the M.S. degree in computer science and technology.