Network Working Group
INTERNET-DRAFT
Expires: May 19, 2008

Ken'ichi Kamada
Shoichi Sakane
Yokogawa Electric Corporation
November 16, 2007

Client-Friendly Cross-Realm Model for Kerberos 5
draft-kamada-krb-client-friendly-cross-03.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft expires on May 19, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document proposes a cross-realm traversal model, which is suitable for resource-limited clients, for Kerberos Version 5. This model relieves the clients of the traversal cost by two means. One moves the cost of consecutive Ticket-Granting Service (TGS) exchanges from clients to Key Distribution Centers (KDCs). The other reduces the traversal cost itself by generating a direct inter-realm relationship between two realms. The document describes behavior of clients and KDCs, but does not specify any wire format, which need to be specified separately.

Table of Contents

1. Introduction

Kerberos Version 5 [RFC4120] has a concept of cross-realm authentication so that principals in different realms can authenticate each other. However in the current cross-realm model, each client has to traverse the authentication path, and the burden of the traversal is not negligible for clients with limited resources, e.g., low computational speed, restricted power consumption [CRPS], or high-latency link to the network.

In the current cross-realm operation, a client obtains a service ticket for a remote principal in the following steps:

1) N TGSes to get cross-realm TGTs in order to traverse the intermediate realms, where N is the number of transit, and

2) One TGS to get the final service ticket.

That is, the client needs to perform N + 1 transactions to obtain a ticket for the remote service.

This document proposes a new cross-realm model, which consists of "dynamic cross mode" and "recursive ticketing mode". The former is intended to reduce transit cost itself, and the latter is to move the cost from clients to KDCs. The document describes behavior of clients and KDCs, but does not specify any wire format, which need to be specified separately.

Terms defined in section 1.7 of RFC 4120 are used throughout this document.

2. Problems on Client Performance

In the current model of cross-realm operation, a client has to transit all realms on the path to the destination realm. When the source realm and the destination realm have a direct inter-realm relationship, a client is able to obtain a service ticket with two TGS transactions (one for a cross-realm TGT and another for the service ticket). When the realms have a multi-hop relationship, a client must transit the intermediate realms before it obtains the service ticket. That is, the client's task increases in proportion to the distance of the relationship.

Two issues can be observed here behind the client load, which are described in the following subsections.

2.1. Long Authentication Path

When a client wants to get a service ticket for a remote realm, it must transit to the remote realm by traversing the intermediate realms on the authentication path to the remote realm. The result of traversal is cached as a cross-realm TGT, but it is nothing more than a per-client optimization. Thus all clients accessing a remote realm must pay the cost separately, even if their resources are limited. For a long authentication path, the cost of the whole system becomes large.

2.2. Client-Centric Ticketing

In Kerberos, any service tickets or cross-realm TGTs are issued via TGS, where a client present a ticket for the TGS and obtains a next ticket. Currently, all TGS transactions are initiated by the client and it needs to get all necessary cross-realm TGTs iteratively before the final service ticket. This process is a burden to a resource-limited client.

3. Proposal of Client-Friendly Cross-Realm Model

In this section, two modes of operation are introduced, Dynamic Cross mode and Recursive Ticketing mode, to solve the issues described in the previous section. These two modes are designed to be independent, that is, can be used separately or in combination.

Dynamic Cross mode solves the issue of the long authentication path. In this mode, if the source realm and the destination realm do not have a direct inter-realm relationship, the source KDC traverses the authentication path by itself, contacts with the remote KDC, and generates a direct inter-realm relationship between them. After that, the source KDC can issue inter-realm TGTs directly for the destination realm. The purpose of this mode is to reduce the traversal cost itself by caching the result of traversal.

Recursive Ticketing mode solves the issue of the client-centric ticketing. Consecutive TGS transactions to get cross-realm TGTs and/or a final service ticket are initiated by a client in the traditional Kerberos, whereas a KDC undertake that process in this mode. The purpose of this mode is to shift the cost of TGSes from a client to a KDC. This does not reduce the cost itself.

3.1. Dynamic Cross Mode

Dynamic Cross mode enables a KDC to issue an inter-realm TGT directly to a remote KDC with which the KDC doesn't preshare an inter-realm

key. To issue an inter-realm TGT directly, a temporary inter-realm key needs to be provided somehow. To achieve that, the local KDC obtains a special ticket for the remote KDC and uses its session key as an inter-realm key. This methodology was introduced by PKCROSS [PKCROSS]. In this document, that special ticket is called as an "inter-KDC ticket", and an inter-realm TGT generated from an inter-KDC ticket is called as a "dynamic inter-realm TGT".

How does the local KDC reach the remote KDC is out of scope of this model, but we can easily come up with 1) traversing a long authentication path if available or 2) using PKINIT. In the context of this model, PKCROSS is interpreted as a combination of this mode and PKINIT.

This document does not standardize a specific protocol, but an inter-KDC ticket will have the following form:

- its sname/realm has a special form "dyncross/REMOTE.REALM@REMOTE.REALM" to indicate that it is a inter-KDC ticket, and

- its cname/crealm is "krbtgt/LOCAL.REALM@LOCAL.REALM".

A dynamic inter-realm TGT will have the following form:

- its TicketExtensions field [KRBEXT] contains the inter-KDC ticket, and

- it is protected by the session key (or the sub-session key) of the inter-KDC ticket.

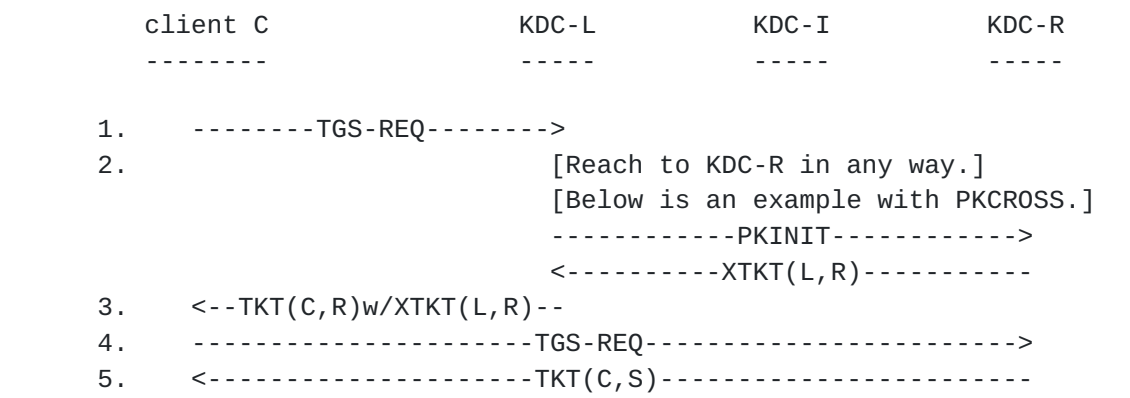


Figure 1: Message Flow of Dynamic Cross Mode

It is critical to verify whether or not the requesting principal is the KDC of the realm when Dynamic Cross mode is used. Thus, when a KDC receives a dynamic inter-realm ticket, it must verify that the inter-KDC ticket's cname/crealm is "krbtgt/REALM@REALM" and the inter-KDC ticket and the dynamic inter-realm ticket has the same crealm.
[[Is this enough?]]
[[Should "pkcross/" allowed?]]

3.2. Recursive Ticketing Mode

Traditionally, a Kerberos client repeats TGS transactions until it gets the final ticket. For example, it has a TGT for its own realm and wants to get a ticket for a service in 3-hop neighbor realm, then it will:

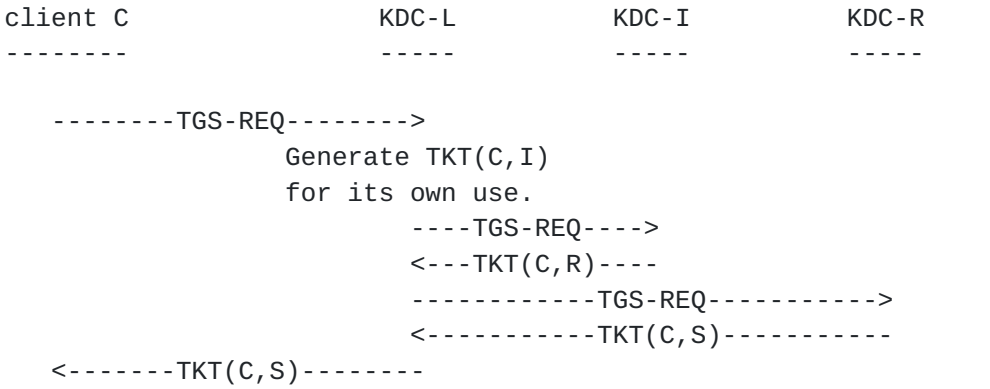
1) Present the TGT and get a cross-realm TGT for the next realm,

2) Present the 1st cross-realm TGT and get a cross-realm TGT for the 2nd next realm,

3) Present the 2nd cross-realm TGT and get a cross-realm TGT for the final realm, and

4) Present the final cross-realm TGT and get a service ticket.

Recursive Ticketing mode enables the client to delegate the KDC to perform all transactions listed above on behalf of the client. An example message flow is shown in Figure 2. The client entrust the KDC with its TGT (step 1). The KDC "impersonates" the client and performs all necessary TGS transactions (steps 2 to 4), and returns the final ticket to the client (step 5).



1. The client C sends a special TGS-REQ, which indicates Recursive Ticketing mode requesting a service ticket for a server S instead of a cross-realm TGT, to KDC-L.
2. KDC-L internally generates a cross-realm TGT that is from C to KDC-I, but does not return it to C.
3. KDC-L uses the generated cross-realm TGT by itself, and sends a TGS-REQ to KDC-I, which requests a cross-realm TGT from C to KDC-R.
4. KDC-L use the obtained cross-realm TGT by itself, and sends a TGS-REQ to KDC-R, which requests a service ticket from C to S.
5. KDC-L returns the final service ticket to C.

Figure 2: Message Flow of Recursive Ticketing Mode

3.3. Combination of the Two Modes

Figure 3 shows a typical message flow when Dynamic Cross mode and Recursive Ticketing mode are used in combination. The figure shows

the case of the initial contact, so a transaction to obtain an inter-KDC ticket is shown (step 2), but it is infrequently used because the XTKT is cached. Usually, only two round-trips do all the work.

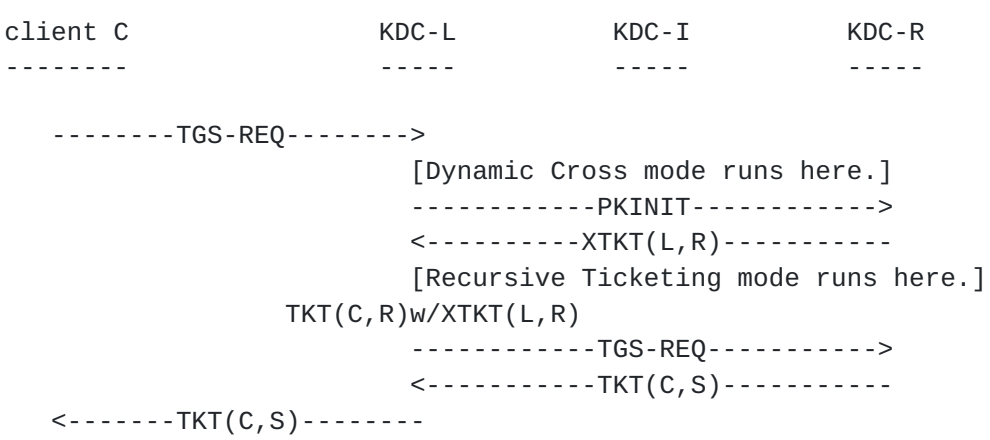


Figure 3: Message Flow When Dynamic Cross Mode and Recursive Ticketing Mode Are Combined

4. Advantage of The Proposed Model for Deployment

4.1. Compatibility with Traditional Kerberos Deployment

Dynamic Cross mode involves only KDCs. From the viewpoint of a client (and a server), it seems that there is a direct inter-realm relationship between two realms. This means that Dynamic Cross mode needs to be deployed only in KDCs. This property is advantageous, because it does not affect large installed base of clients. One impeding factor in practice is that some existing implementations cannot handle ticket extensions transparently. This is further discussed in Interoperability Considerations section.

Recursive Ticketing mode involves only a client and its local KDC. From the viewpoint of the remote KDC, TGS-REQs from a KDC in recursive mode cannot be distinguished from those from a "genuine" client (except caddr; see Interoperability Considerations section). Resulting service ticket is identical to the traditional one, so the remote server has nothing to do with this mode. In short, Recursive Ticketing mode can be deployed in local realm, independently of the remote deployment. The merit of this property is large, because remote realms are often in different administration.

4.2. Orthogonality of the Two Modes

Dynamic Cross mode and Recursive Ticketing mode are independent concepts. Both can be implemented separately or can be used in

combination. When they are combined, the load of clients are shifted to KDCs and additional load of KDCs are minimized, thus efficient cross-realm environment is achieved.

5. Front-End Protocol for Recursive Ticketing Mode

This document does not specify wire-level protocol, which will be done in another document. This section provides some candidates for the protocol, which is used to request Recursive Ticketing mode from a KDC (Figure 4). This protocol is hereinafter called as Attorney Request. Attorney Request is effective only in TGS-REQ.

When a KDC receives Attorney Request, it can choose another method than Recursive Ticketing mode, as long as the KDC's behavior for clients is identical to the mode. The inter-TGS protocol (XTGSP) [XTGSP] is an example of this.

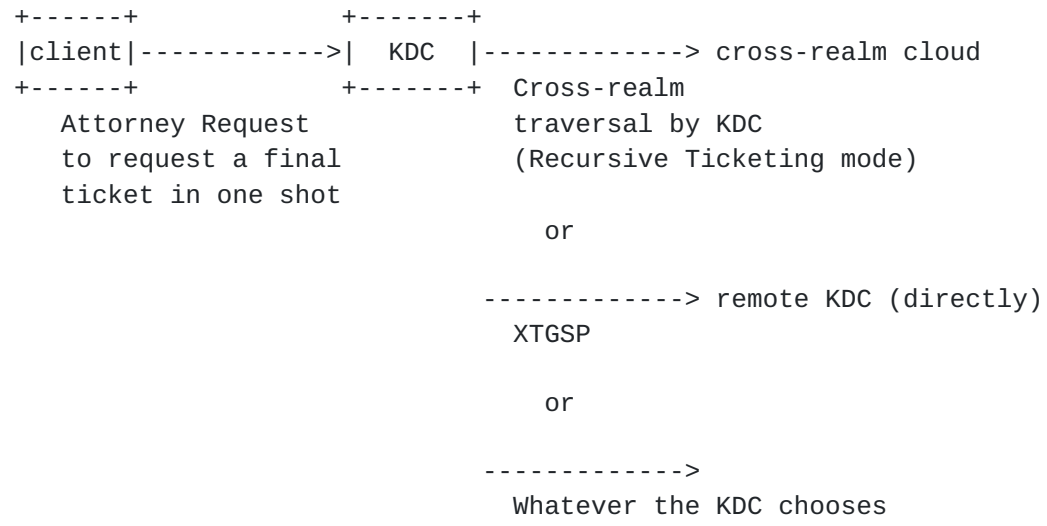


Figure 4: Front-End Protocol for Recursive Ticketing Mode

Candidate 1: Implicit Signaling

A client simply requests a final ticket to the local KDC. If the KDC supports this implicit protocol, it will process the request. If not, KDC_ERR_S_PRINCIPAL_UNKNOWN will be returned. A possible drawback is that if a requested final ticket is for a TGS, the KDC does not know whether the client expects normal mode or Recursive Ticketing mode. In addition, implicit signaling can conflict with future extensions.

Candidate 2: Explicit Signaling

Define "attorney" flag in KDCOptions or a pre-authentication type to request Recursive Ticketing mode.
[[what happens if not supported?]]

6. Related Protocols Currently Proposed

6.1. PKCROSS

PKCROSS will be usable as a protocol for Dynamic Cross mode.

It should be noted that the requesting principal must be verified as described in section 3.1. However, in the case of PKCROSS, the client's realm name is not available because a PKCROSS request is an AS-REQ, which does not have the crealm field.
[[So, what name should the local KDC use?]]

6.2. XTGSP

The purpose of XTGS protocol is similar to that of this model, but the behavior is somewhat different [XTGSP]. If XTGS is viewed from the perspective of this model, it blends the two modes indivisibly to reduce the number of messages between KDCs as far as possible at the price of the abstraction of cross-realm TGTs and inter-KDC tickets.

Once Attorney Request protocol is standardized, XTGS can be used as an opaque back-end.

7. Interoperability Considerations

User-to-user mode
Attorney Request protocol should be able to indicate user-to-user authentication.

The addresses field in TGS-REQ
This field is copied into the caddr field in EncTicketPart, so if this field is used in a TGS-REQ, the resulting ticket can be used only from the specified addresses. When the local KDC receives an Attorney Request and decided to go Recursive Ticketing mode, it should copy the addresses field only into the final TGS-REQ in the recursive process. It must not copy the field into TGS-REQs to intermediate KDCs, because resulting tickets are to be used by the local KDC instead of the client.

Opacity of ticket extensions
The ticket extensions defined in rfc1510ter [KRBEXT] extends the Ticket ASN.1 type, which is visible to clients. This is not a

problem if a client implementation treats a Ticket as an opaque data, and there are such implementations, but unfortunately the major free implementations do not. On the other hand, there is a proposal of etype-based ticket extensions [TKTEXTALT]. It encapsulates cleartext bits in the enc-part component of a Ticket. It should not have any problems of opacity.

[[negotiation of various parameters]]

[[If there are multiple authentication paths and a client has enough knowledge, it could choose which path to take. With Recursive Ticketing mode, it cannot because it is up to the KDC to select the path. Is this a problem? With Dynamic Cross mode, it can as before.]]

[[co-existence with the plain Kerberos; Attorney requesting client vs. non-attorney KDC; inter-realm generating local KDC vs. non-generating remote KDC]]

[[anything to do with referral?]]

[[when a KDC in Recursive Ticketing mode receives a KRB-ERROR?]]

8. Security Considerations

8.1. Denial of Service (DoS)

A KDC that implements Recursive Ticketing mode needs to initiate multiple TGS-REQs upon a request from a client. This means that the KDC will have some states in it and may suffer from DoS attacks.

Fortunately, Recursive Ticketing mode can be requested in TGS-REQ, which is only available to authenticated clients, thus, any untrusted party cannot exploit this statefulness.

8.2. Ticketing Policy

Recursive Ticketing mode changes nothing about the messages sent to the intermediate and remote KDCs. Those KDCs will not notice the difference and their ticketing process have nothing to be changed.

Dynamic Cross mode dynamically generates new authentication paths. This means that KDCs that are involved in the transit of a client are different from those that would be involved if this mode were not used.

- Parameters of cross-realm TGTs (lifetime and flags) for a new relationship need to be dynamically transferred (a la PKCROSS).

- How to handle the transited fields in inter-KDC tickets, dynamic inter-realm tickets, and service tickets?

- Where the remote KDC adds AuthorizationData and the end-server checks it: there is no problem because it is a local matter of the remote realm.

- Where an intermediate KDC adds AuthorizationData: traditionally it is added in a cross-realm TGT and propagated to the service ticket; now it will be propagated to the inter-KDC ticket. Should AuthorizationData in an inter-KDC ticket be copied into a cross-realm TGT or not? Even if it is copied, AuthorizationData on inter-KDC ticket cannot represent per-client information, so if it is necessary, Dynamic Cross mode must not be used.

8.3. Authorization of Client KDCs in Dynamic Cross Mode

Dynamic Cross mode issues a XTKT, which is a service ticket to use "dyncross" (or "pkcross") service. This ticket is used to build dynamic inter-realm TGTs, so a principal that possesses it can act as a KDC. Thus it must not be used by arbitrary clients except the genuine KDCs. In other words, "dyncross" (or "pkcross") service requires authorization.

PKCROSS document does not specify how to authorize the requesting principal. Considering section 3.2 of PKINIT [RFC4556], id-pkinit-KPKdc should be checked, but this information is available only when issuing XTKTs and not when verifying the XTKTs. Two possible ways to circumvent to this are listed below.

- To put AuthorizationData and the information of EKU in it.

- To authorize a client when issuing an XTKT, though this behavior is different from the normal model of Kerberos.

9. IANA Considerations

This document has no actions for IANA.

10. Acknowledgments

The authors would like to acknowledge Saber Zrelli, Masahiro Ishiyama, Atsushi Inoue, Kazunori Miyazawa, and Nobuo Okabe for

contributions to this document.

11. References

11.1. Normative References

[KRBEXT] Yu, T., "The Kerberos Network Authentication Service (Version 5)", draft-ietf-krb-wg-rfc1510ter-04, Work in Progress, March 2007.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

11.2. Informative References

[CRPS] Sakane, S. et al., "Problem statement on the cross-realm operation of Kerberos", draft-ietf-krb-wg-cross-problem-statement-00, Work in Progress, September 2007.

[PKCROSS] Hur, M. et al., "Public Key Cryptography for Cross-Realm Authentication in Kerberos", draft-ietf-cat-kerberos-pk-cross-08, Work in Progress, November 2001.

[RFC4556] Zhu, L., B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, June 2006.

[TKTEXTALT] Message-ID: <tslfy54akcb.fsf@mit.edu>.

[XTGSP] Zrelli, S. et al., "XTGSP, the Inter-TGS protocol for cross-realm operations in Kerberos", draft-zrelli-krb-xtgsp-01, Work in Progress, March 2007.

Authors' Addresses

Ken'ichi Kamada
Shoichi Sakane
Yokogawa Electric Corporation
2-9-32 Nakacho, Musashino-shi,
Tokyo 180-8750 Japan
E-mail: Ken-ichi.Kamada@jp.yokogawa.com,
Shouichi.Sakane@jp.yokogawa.com

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.