

Specifying Kerberos 5 Cross-Realm Authentication

Chris Walstad

Joint work with Iliano Cervesato,
Aaron D. Jaggard, Andre Scedrov

Supported by ONR, NSF, NRL

Overview

- Introduction
- Kerberos 5
- Formalization
- Properties
- Vulnerabilities

Overview of Results

- Formalize cross-realm authentication in Kerberos 5
 - Use MSR
- Adapt Dolev-Yao intruder to cross-realm setting
- Prove property of a critical field in cross-realm ticket
- Highlight vulnerabilities in the presence of compromised intermediate realms
 - Kerberos specifications disclaim responsibility for these

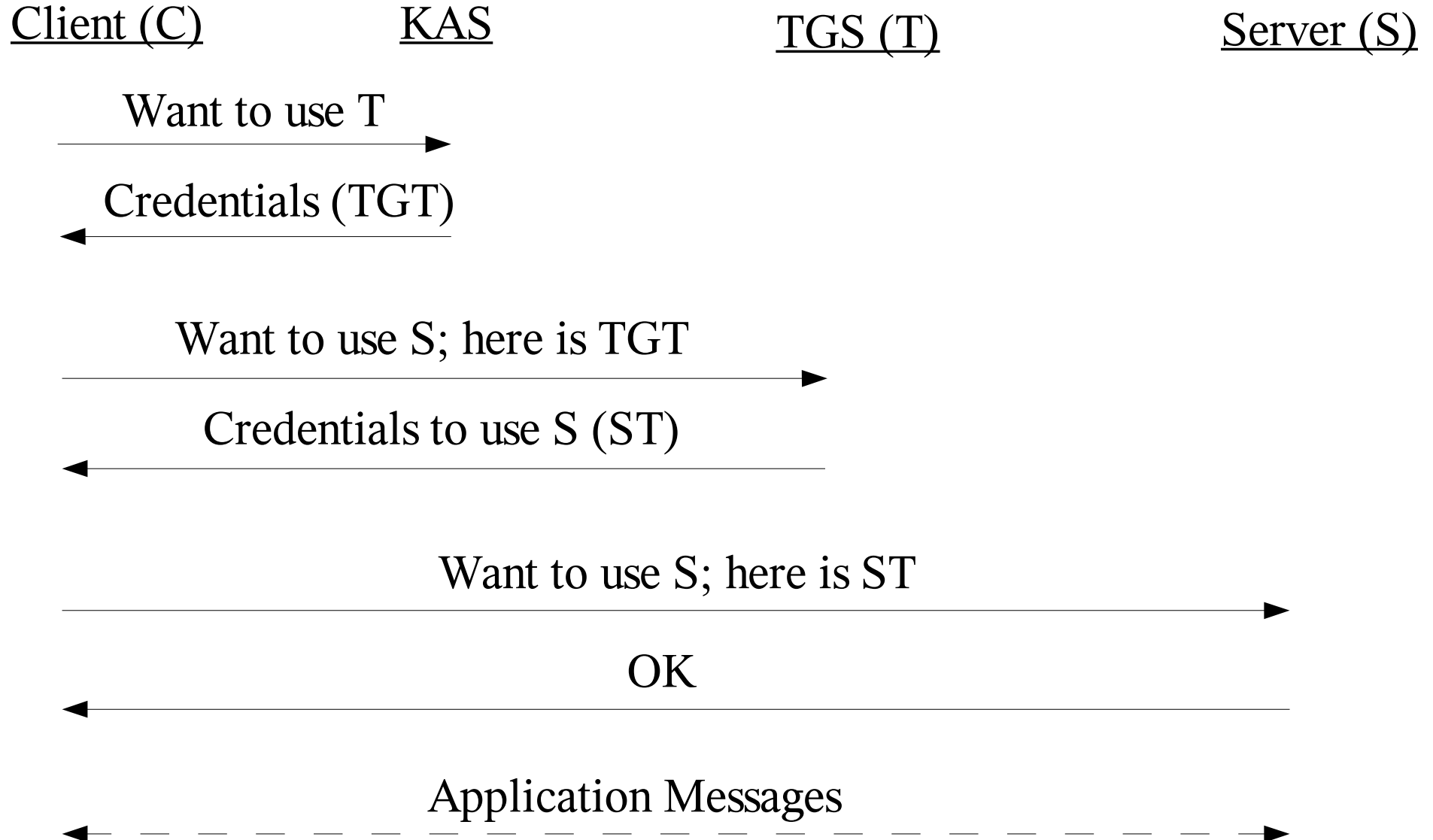
Background and Related Work

- Kerberos – intra-realm has been extensively studied
 - Kerberos 4 analyzed using inductive approach (Bella & Paulson)
 - Kerberos 5
 - Simplified version analysed with Mur ϕ (Mitchell, Mitchell, & Stern)
 - Detailed formalization of intra-realm authentication analyzed using MSR (Butler, Cervesato, Jaggard, Scedrov)
 - Current project is a continuation of this work
- Cross-realm authentication
 - Hierarchical organization of authentication servers (Birrell *et al.*)
 - Similar to natural organization for Kerberos
 - Define local trust policies that mitigate global security exposure (Gligor *et al.*)

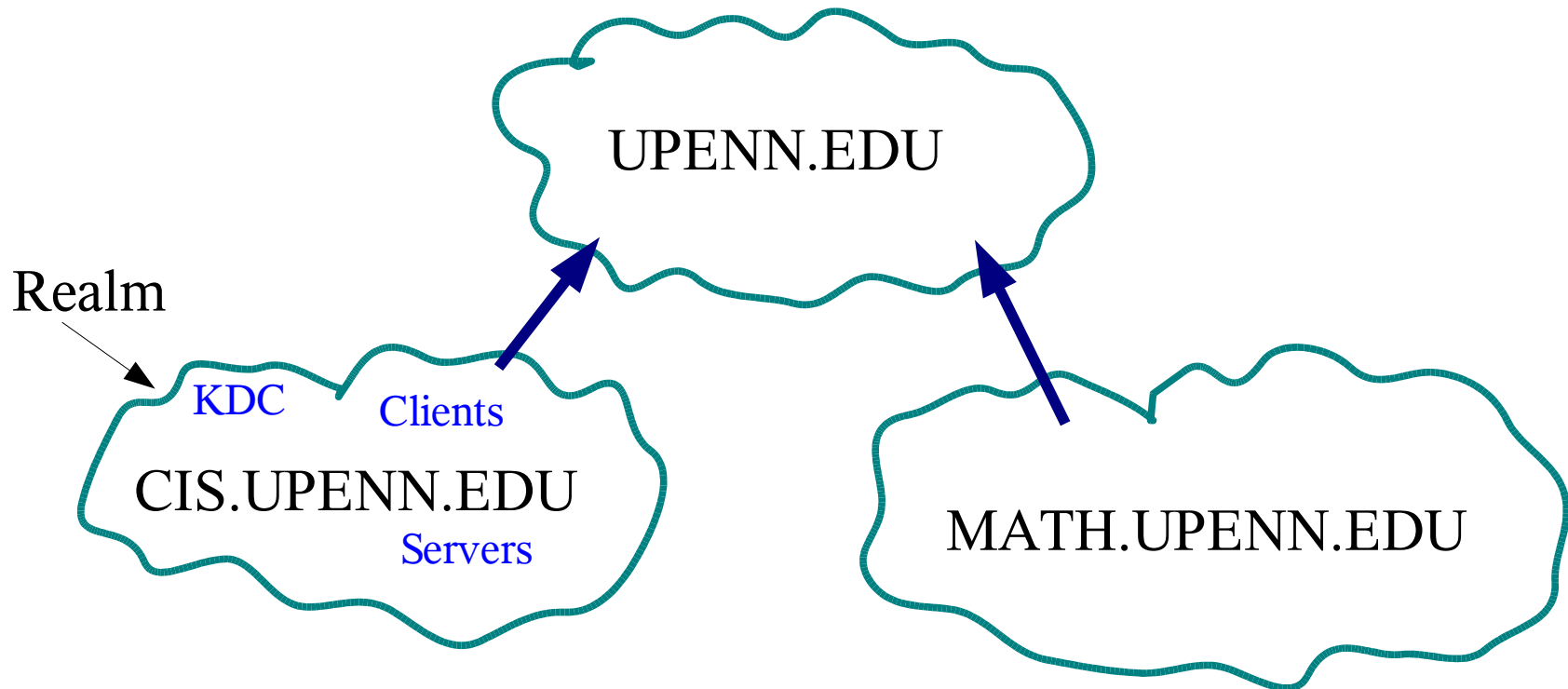
Kerberos 5

- Authentication
 - Single sign-on
 - Repeatedly authenticate a client to multiple servers
- Authentication Server (**KAS**)
 - Provides long term (*e.g.*, 1 day) ticket called a **Ticket Granting Ticket** (TGT)
 - Uses client's long term key (*e.g.*, derived from password)
- Ticket Granting Server (**TGS**)
 - Provides short term (*e.g.*, 5 minutes) ticket called **Service Ticket** (ST) based on client's TGT
 - Client uses ST to access the server

Intra-Realm Messages



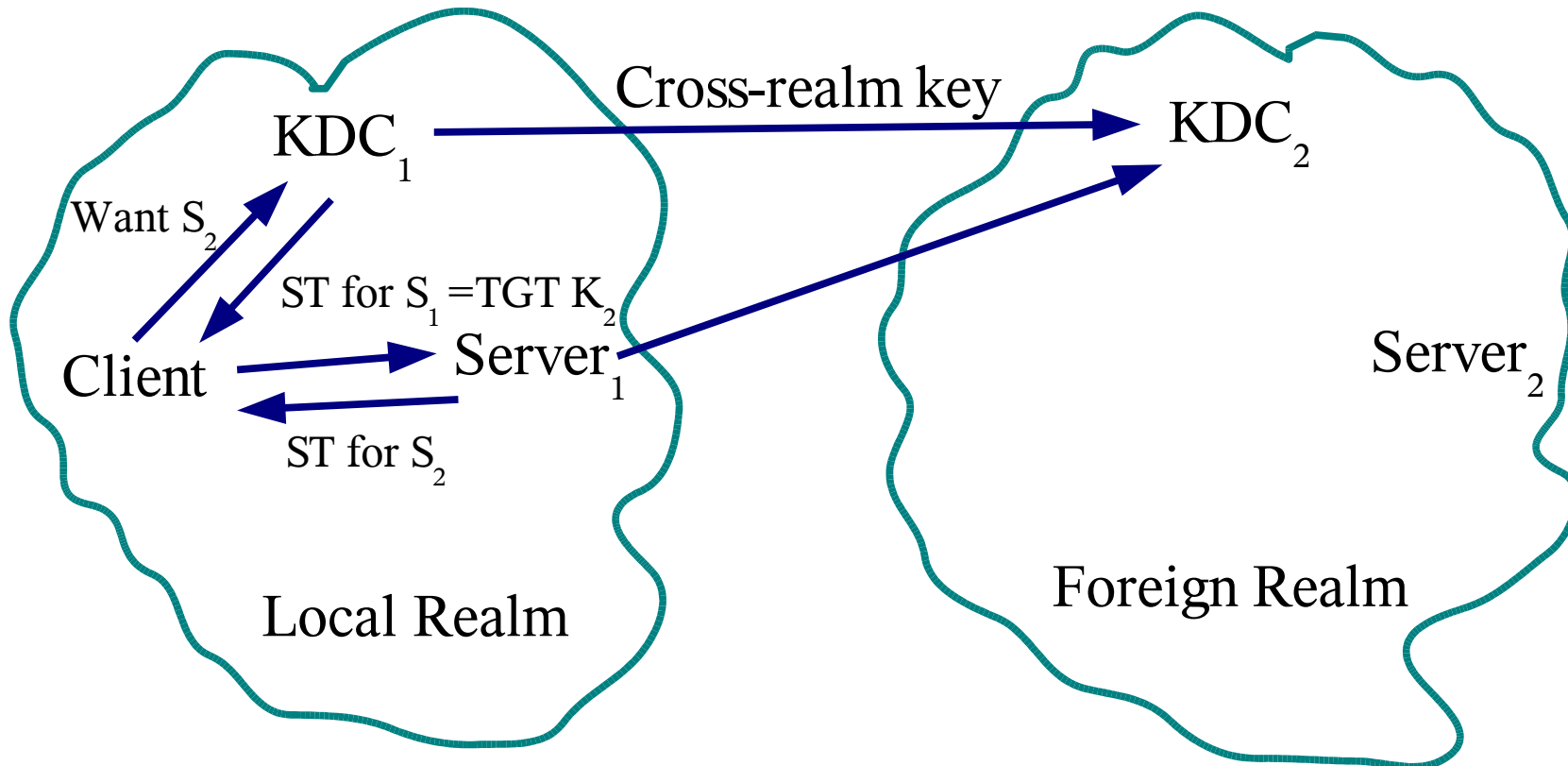
Cross-Realm Kerberos 5



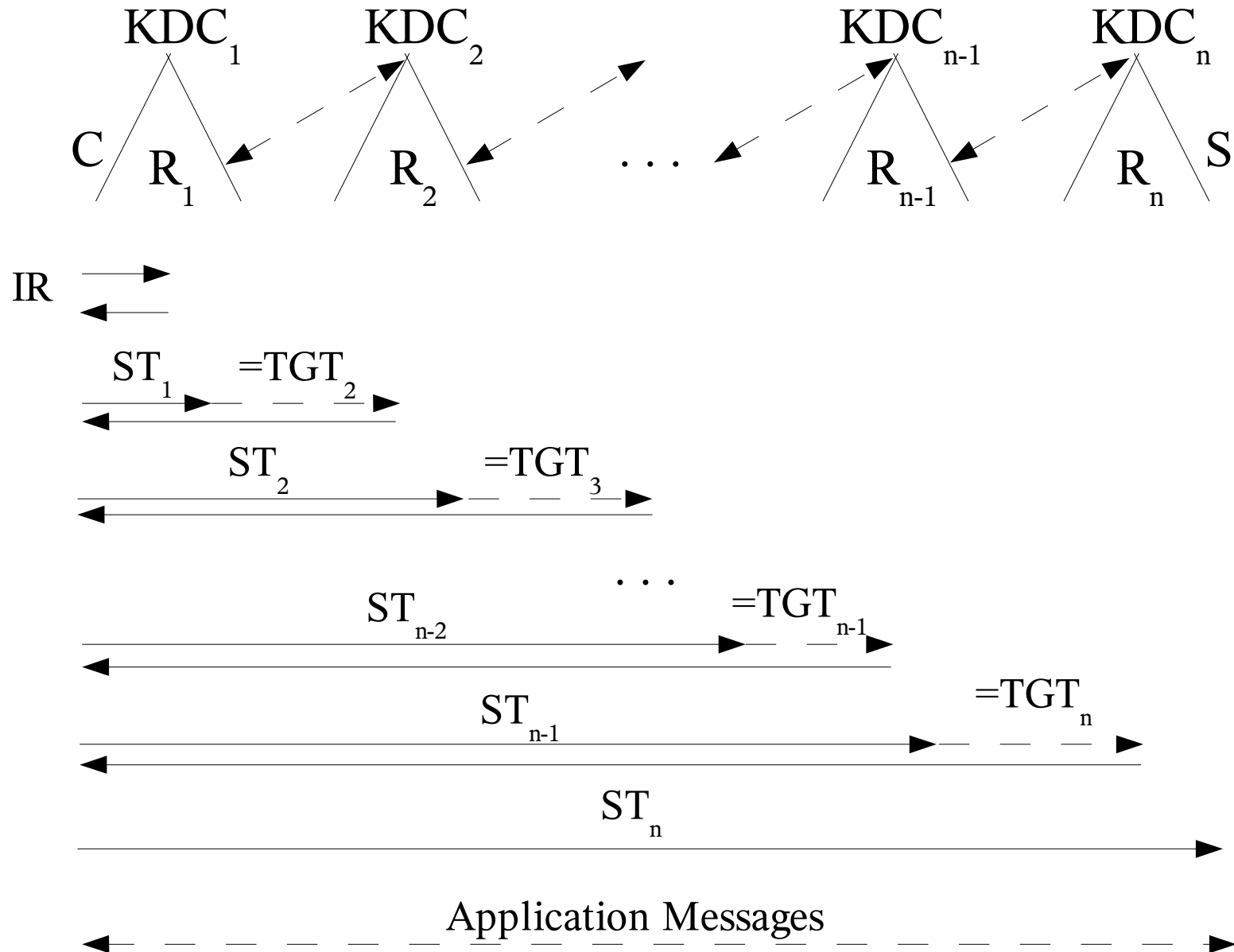
- Authenticate clients across organizational boundaries
 - Simpler administration
 - Better user experience

Cross-Realm Kerberos 5

- Register KDC of foreign realm as a server in local realm
 - Cross-realm key
 - Service ticket for foreign KDC is interpreted as a TGT



Cross-Realm Messages



Cross-Realm Kerberos 5

- Recommended organization of realms is hierarchical
 - “Shortcuts” allowed
- **Authentication path** is path through traversed realms
 - TGS adds previous realm name to **TRANSITED** field

Formalization

- Use MSR 2.0 (Cervesato)
- Models both intra- and cross-realm authentication
 - Is a continuation of prior work done on intra-realm authentication
- Includes the minimum level of detail we believe necessary to prove properties on authentication, confidentiality, and the effect of compromised realms
- Validation using MSR Implementation developed by Cervesato, Reich, and Stehr underway

Formalization

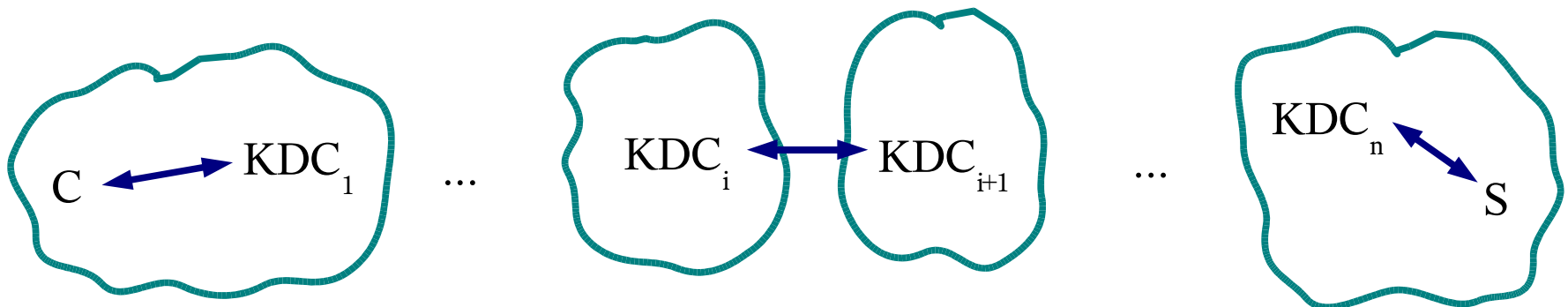
- Realm type
 - Each principal is parameterized by realm it lives in
- Database keys modified to handle cross-realm keys
- r TGS allows us to view this principal as an application server in one realm and a TGS in another realm
- Support for TRANSITED field
- Rule for TGS returning a cross-realm ticket
- Existing rules and types updated

Intruder Model

- Intra-realm setting: unavoidable assumption is that the KAS and TGS behave honestly
- Cross-realm setting: must consider compromised remote KDC
 - Local system administrator has no control over other realms
 - How can a compromised remote KDC affect the rest of the Kerberized network?
- If a realm is compromised then the intruder possesses all of the database (long-term) keys
- Assume a worst-case scenario in which all principals communicate on the same network

Theorem

- If there are any compromised realms involved in authentication then at least one of them will appear in the TRANSITED field
- If invalid/improper authentication took place then the intruder possessed one of the following keys
 - The key shared by the end-server and the TGS of that realm
 - A cross-realm key for some pair of TGSs on the authentication path
 - The client's long-term secret key



Proof Methods

- Rank and corank functions
 - Inspired by Schneider's rank functions in CSP
 - k-Rank - work done using key k (data origin authentication)
 - E-Corank - work needed using keys from E (secrecy)
- Valid credential presented to S has positive rank
 - No MSR facts of positive rank at start of protocol run
 - Examine principal and intruder rules
 - Which keys must be lost to allow intruder to increase rank?
 - Which honest principals can increase this rank?

Proof Outline

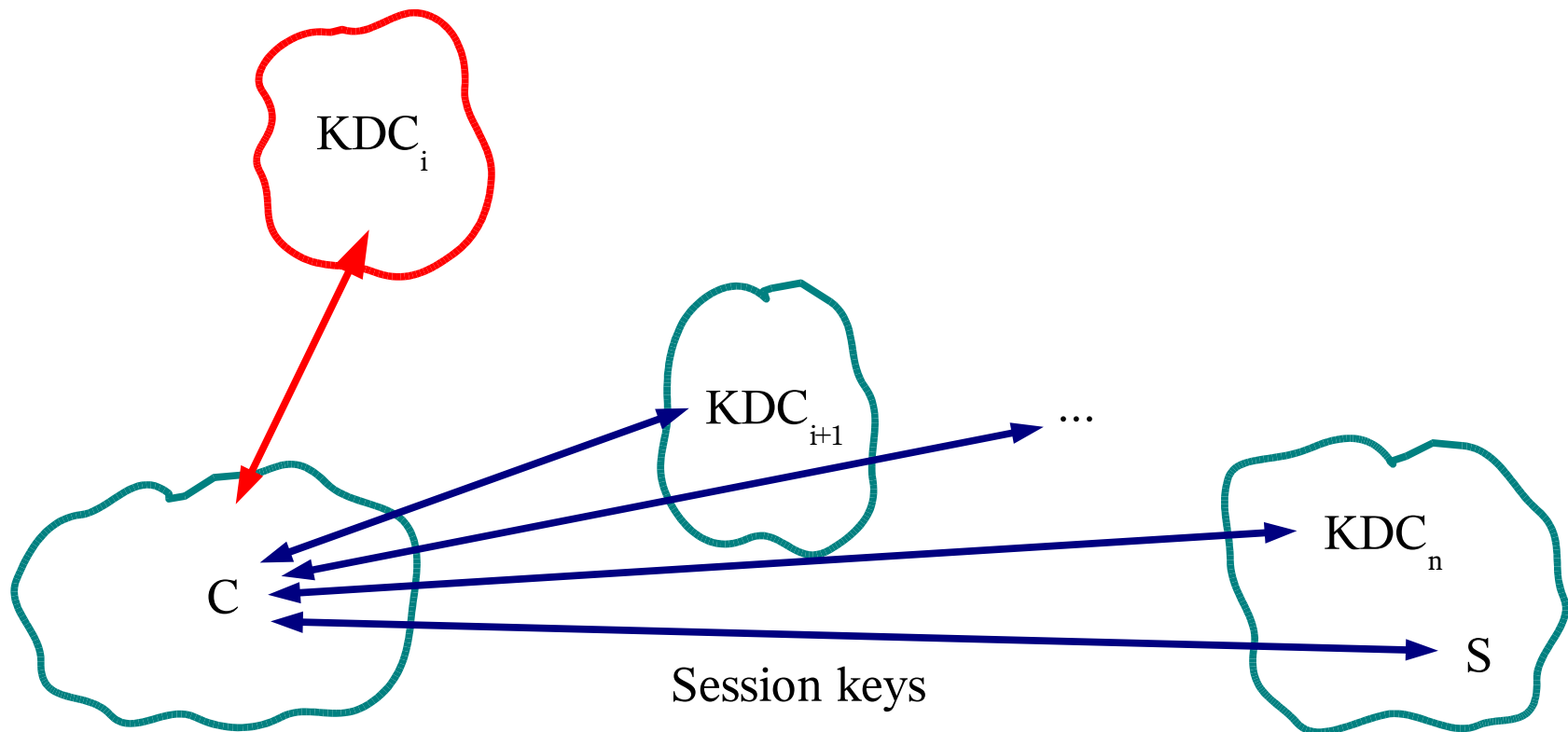
- S sees a valid request (a fact with positive rank)
- If honest principal created this fact, he did so after seeing a fact of (a different) positive rank
 - Every honest TGS adds name of preceding realm
- If intruder created this fact, then the key used has been lost to the intruder
- Chain this argument backwards (reaching a fact created by the intruder or C's initial request---no facts of positive rank initially, so this process must stop)
 - If intruder created some fact in the chain, then some key has been lost; the compromised realm is named in the request
 - If the intruder did not, then exactly the transited realms appear in the transited field

Vulnerabilities

- Kerberos specifications make no guarantees if a trusted foreign realm becomes compromised
 - Therefore these vulnerabilities are not attacks and the specification disclaims responsibility for them
- System administrators should be made aware of exactly what damage can be done by a compromised foreign realm
- Identified 3 vulnerabilities
 - Not necessarily complete

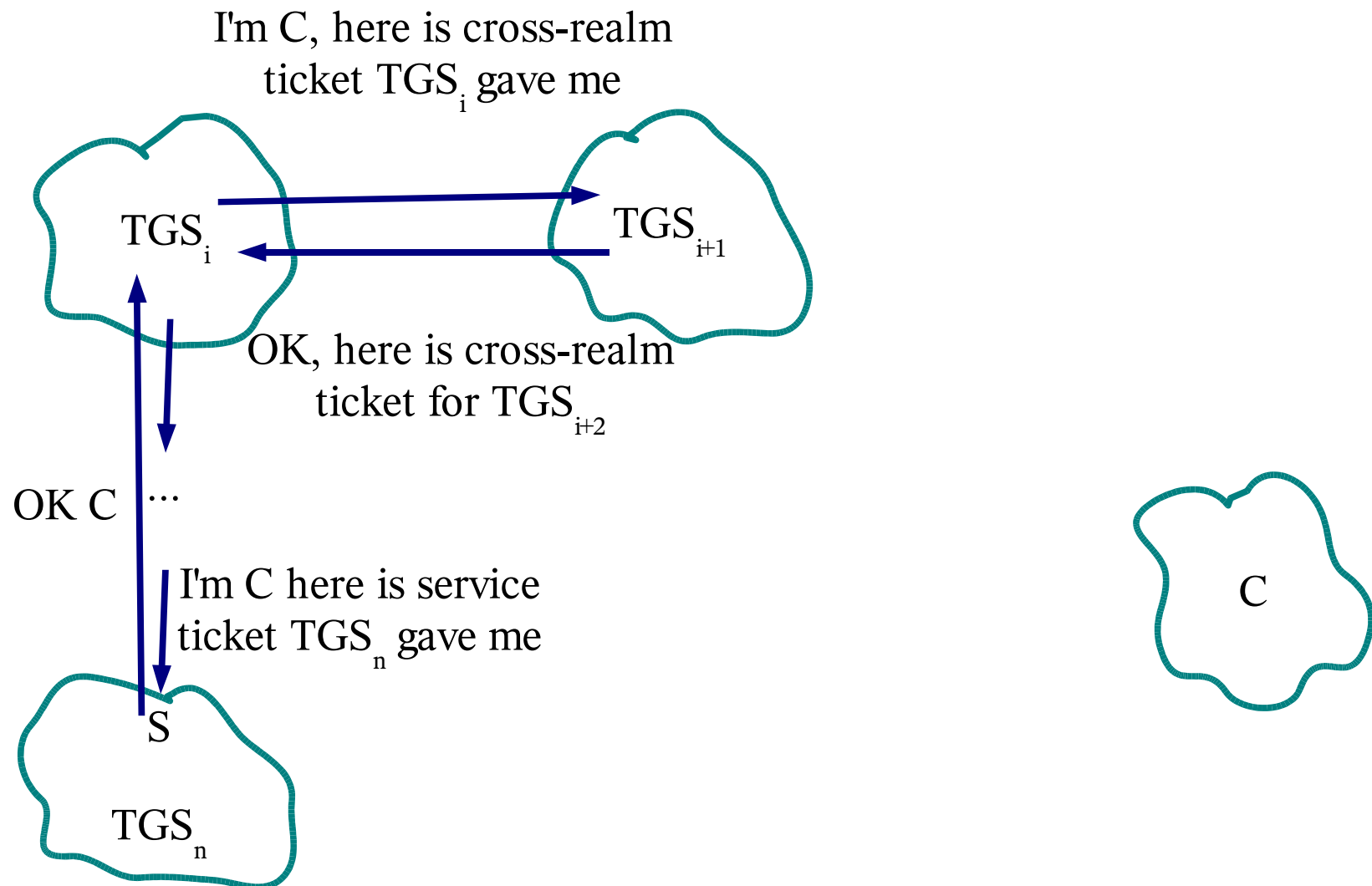
Vulnerabilities

- All TGSs on the authentication path are capable of learning the key shared between the server and the client as well as all of the session keys shared by the client and each TGS on the authentication path



Vulnerabilities

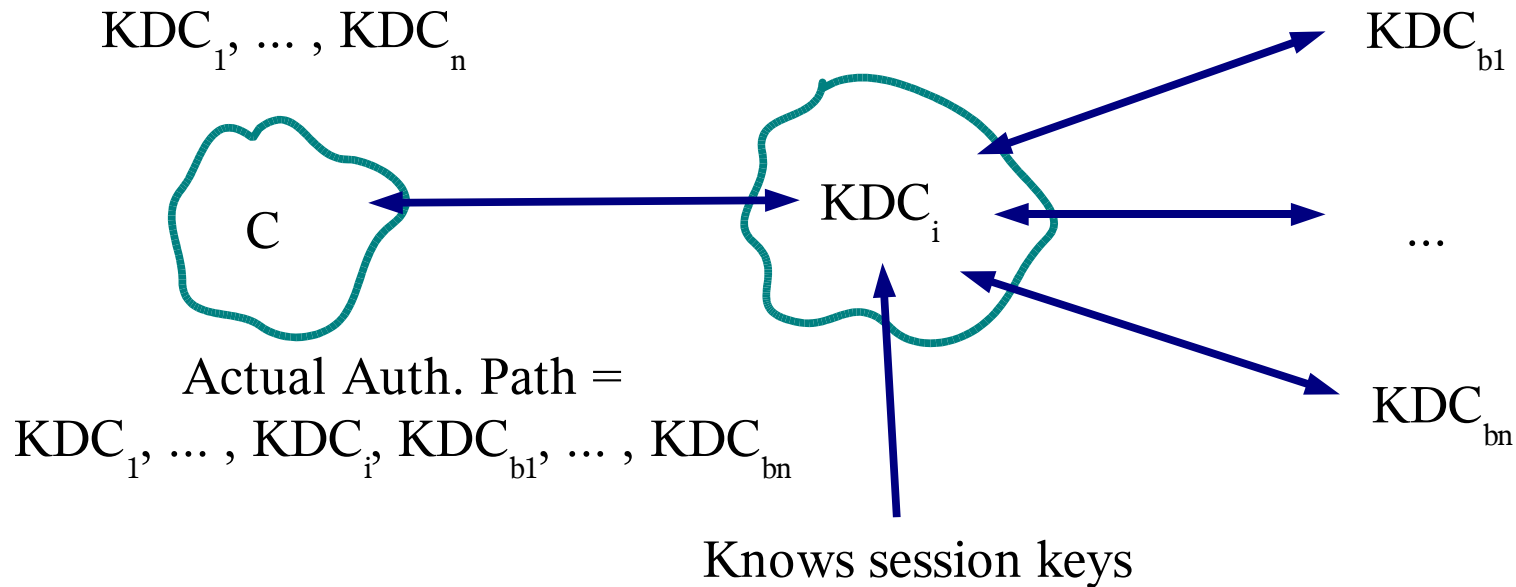
- Remote TGS can impersonate a client anywhere outside of the client's realm



Vulnerabilities

- If there is a compromised KDC on the authentication path then that KDC can trick the client into believing she is following a false authentication path

C believes Auth. Path =
 KDC_1, \dots, KDC_n



Conclusions

- Formalized cross-realm authentication in Kerberos 5
- Extended Dolev-Yao intruder
- Characterized minimum requirements in view of assessing confidentiality and authentication properties
- Documented a range of harmful behaviors

Future Work

- Prove traditional confidentiality and authentication properties
- Analyze PKINIT and PKCROSS subprotocols that may help mitigate the harm that a compromised KDC can inflict

Sample Rule

$\forall C: \text{client } R_C$

$\forall X: \text{msg}$

$\forall T: \text{TGS } R_T$

$\forall T_n: \text{ts } R_n$

$\forall AK: \text{shK } C \ T$

$\forall t_C: \text{time}$

$\exists n_2: \text{nonce}$

$\cdot \Rightarrow$

$N(X, \{C, t_C\}_{AK}, C, T_n, n_2)$

$L(C, T_n, T, AK, n_2)$

if $\text{Auth}_C(X, T, AK), \text{DesiredHop}(C, T_n, R_T, R_n), \text{clock}_C(t_C)$

- This is part of the client's role in TGS exchange

Intruder Model

- Assume a worst-case scenario in which all principals communicate on the same network
- Single Dolev-Yao intruder that can impersonate clients, end-servers, and KDCs

$\forall R : \text{realm} \quad \forall P : \text{tcs } R \quad \forall k : \text{dbK}^R P$

$\cdot \Rightarrow I(k)$

if compromised(R)