University of Parma
Department of Information Engineering
Parma, Italy

A Resilient Architecture for DHT-based Distributed Collaborative Environments

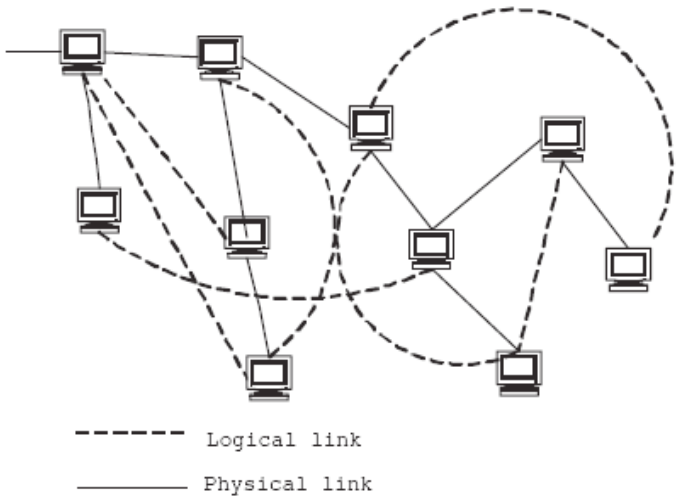
Simone Cirani, Natalya Fedotova, Luca Veltri

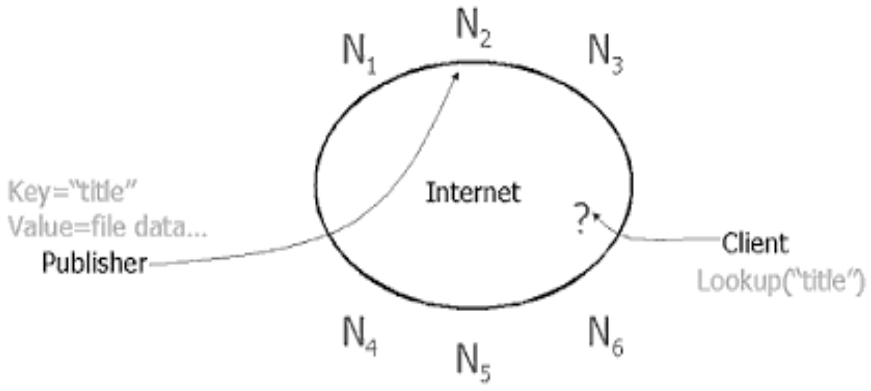
International RISE/EFTS joint workshop on Software Engineering for Resilient Systems (**SERENE-2008**),

Newcastle upon Tyne (UK), November 17-19, 2008

DHT-based P2P networks

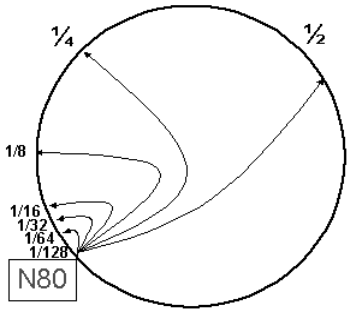
- Currently, Distributed Hash Tables (DHT) are widely used by a number of peer-to-peer (P2P) applications as data storage and retrieval infrastructure (eDonkey, BitTorrent, CFS, OceanStore, etc)
- DHT-mechanisms are realized on the overlay layer through building up a virtual communication scheme above the existent network topology
- The routing table-based look-up service maps a given key (identifier) to a node that is responsible for the key using a hash function
- The mapping rules and look-up metrics are defined by a specific routing algorithm applied (Chord, Kademlia, Pastry, Tapestry, etc)





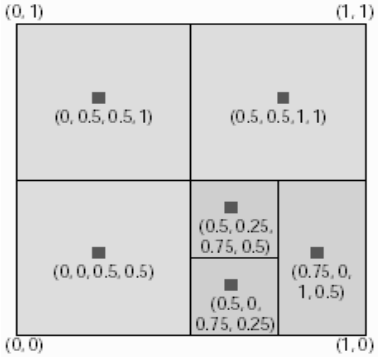
DHT look-up algorithms: data structure of routing tables

SKIP-LIST



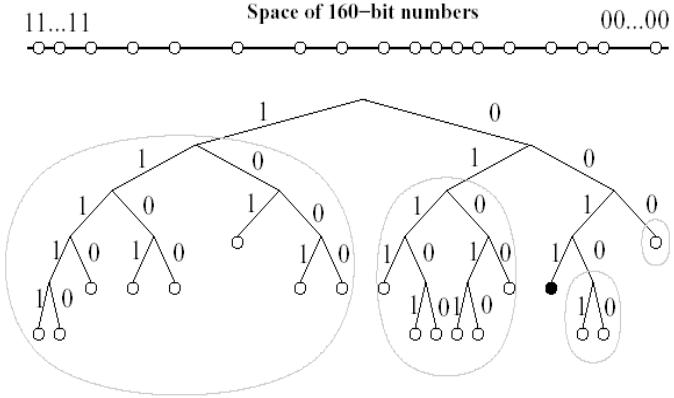
Chord

RECTANGLES

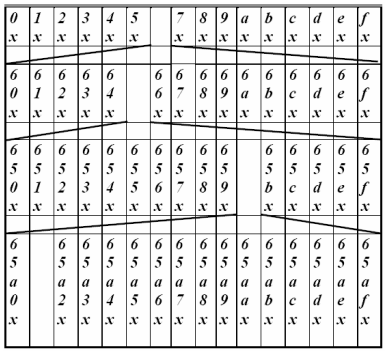


CAN

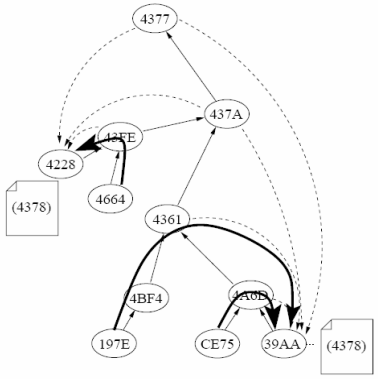
TREE



Kademlia



Pastry



Tapestry

DHT mechanisms

- Advantages:
  - Efficient routing performance
  - Self-organizing data storage and lookup service
  - High scalability
  - Automatic load balancing
  - No single point of failure
  - Simple API

Resilience through DHT mechanisms

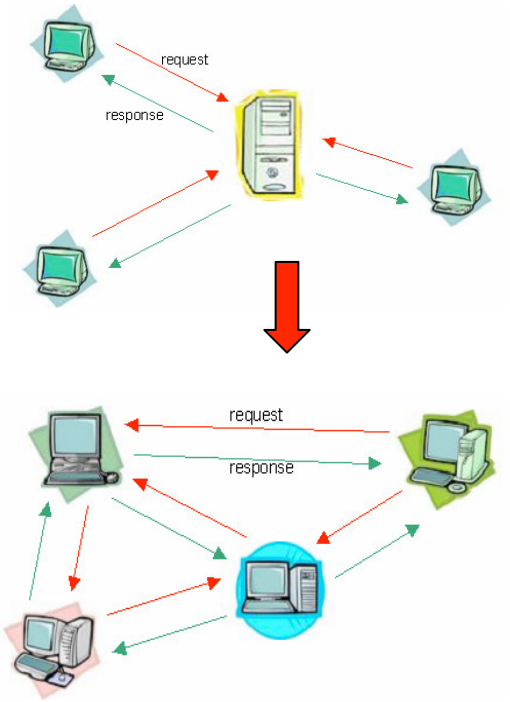
DHT-based algorithm	Factors critical for system resilience	Resilience mechanisms
Chord	*a node's knowledge of its predecessor and successor in the identifier space is indispensable, i.e. each node's finger lists should be correctly maintained and* ***perfectly updated*** *in order to reflect any single procedure within the DHT.*	*key redundancy*
Pastry	*the stability of leaf sets of single peers (lists of the closest nodes) is important, so when some node goes off-line, leaf sets of its neighbors should be* ***immediately updated***	*key redundancy*
Kademlia	*the* ***last-recently seen eviction policy*** *is applied to minimize changes in the population of k-buckets and to maintain contacts with the* ***highest uptime****.* *Maintaining lists of the closest peers* ***is not requested****.*	*key redundancy,* *storage redundancy*

Related work: cooperative storage and data management

Application	DHT algorithm applied	Data management scheme	Access permission level
CFS	*Chord*	*block-level storage through multiple replication*	*read-only*
OverCite	*not specified*	*meta-data-based data storage through partitioning the inverted index among peers*	*read-only*
OceanStore	*Tapestry*	*persistent storage through surrogate prefix-matching routing scheme with redundant paths*	*read and write (if authorized by "primary tier")*

Previous work

- extending the limits of application of DHT mechanisms from data storage and retrieval in distributed computer systems to collaborative computing context
- introducing a distributed P2P data sharing system based on Kademlia DHTs into the enterprise environment to organize data management and cooperative work between geographically distant users engaged in a common task
- realization of a distributed work-sharing system which allows data produced by some department to be maximally available for other interested entities in accordance with a predefined policy of attribution of different data access privileges to different users

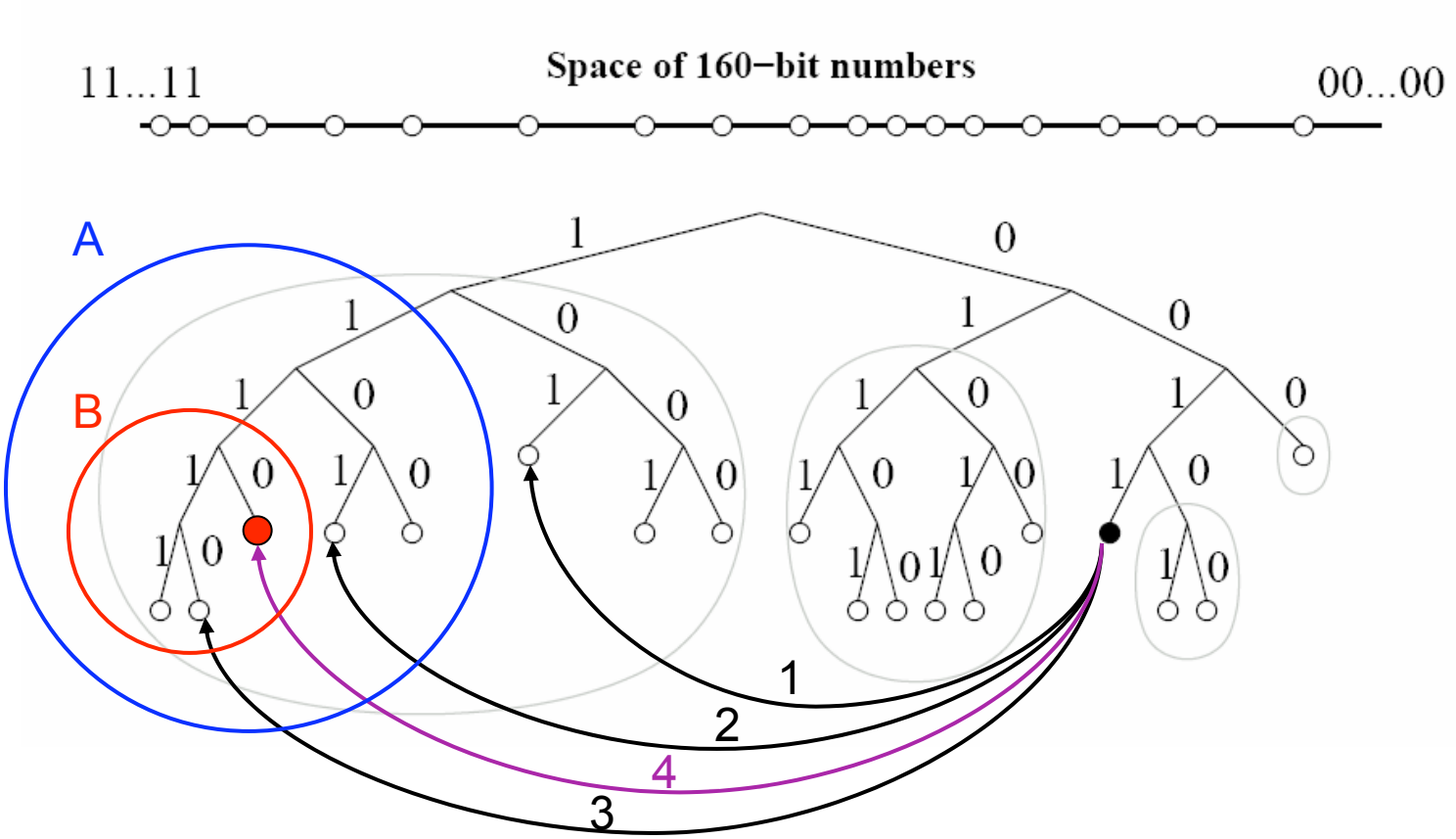


Obtained advantages

- *system exploits hardware and memory resources of all network terminals*
- *high scalability: possibility of incremental growth of the network, delivering complementary capacity when and where needed*
- *no single point of failure*
- *no DoS*
- *no "packet filtering"*

Kademlia

- provides a tree-like data structure of routing scheme, where network nodes are considered as “leaves” of a binary tree
- all nodes and resources have 160-bit identifiers (keys)
- distance between two identifiers: *d (x,y) = x XOR y*
- k-bucket: *<IPaddress,UDPport,nodeID> for nodes from* $d \in \left[2^i ; 2^{i+1}\right)$ for each *0≤i<160* sorted by time last seen

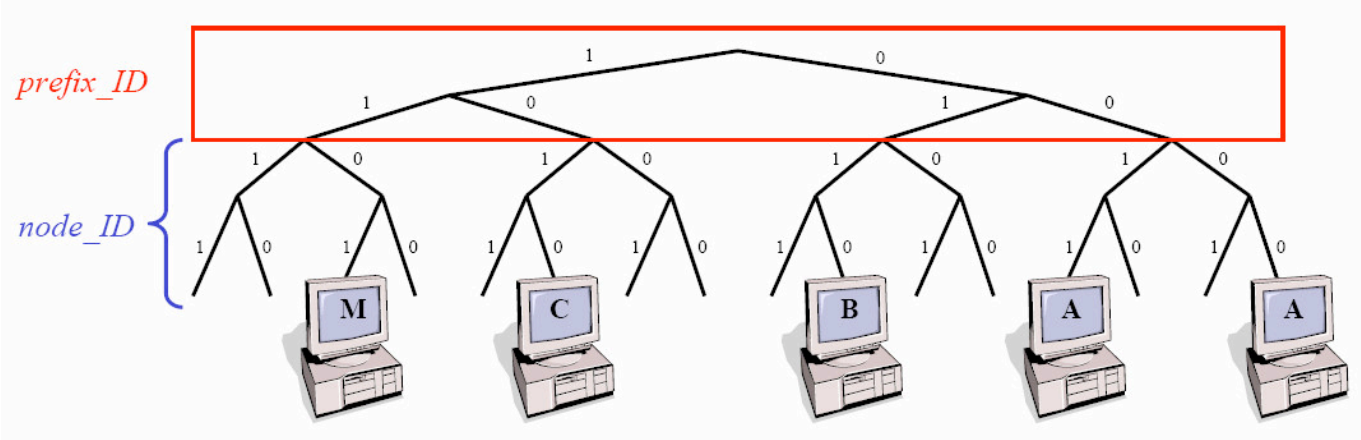


Previous work: proposed solution

- a network community is represented by several groups of nodes with different levels (1, 2, 3…) of responsibilities and data access rights
- a new identification system based on use of prefix identifiers for both nodes and resources is introduced to handle read/write permissions in accordance with a certain enterprise hierarchical model

  ID = <*prefix_ID ; node_ID*>

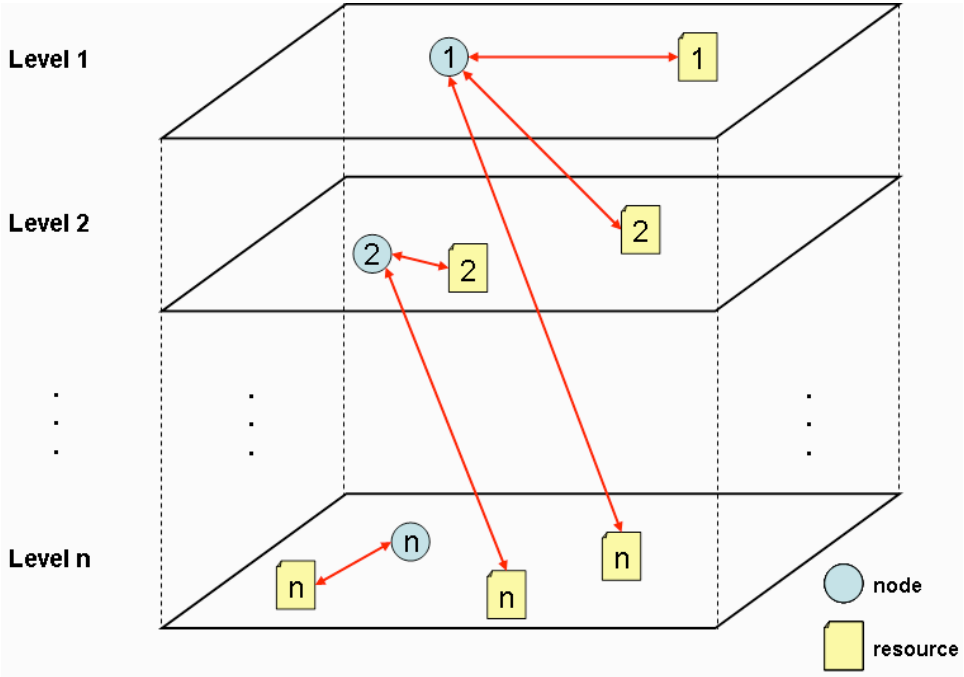
  Key = <*prefix_ID ; key_ID*>
- the prefix indicates the level of “confidentiality” of a file’s content, i.e. if it is for common use or only for limited use of certain work-groups.
- common use resources are hosted by nodes of the lowest level



Strict scheme for hierarchical resource management

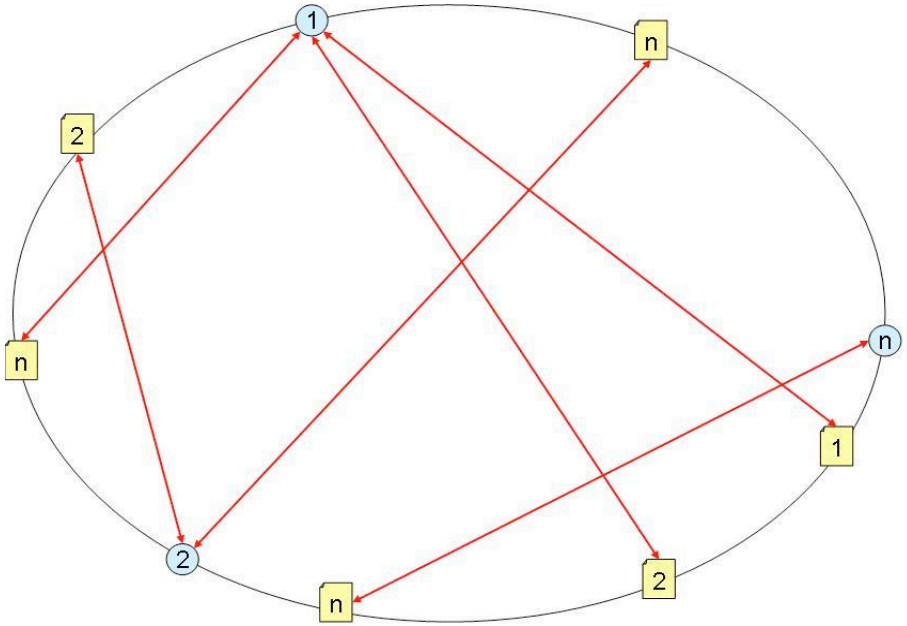
The proposed data management scheme supposes a **strong correspondence** between the level of data confidentiality (represented by the prefix ID) and their actual location (identified by the resource key), where ***β*** bits out of ***n*** bits of the ID are used for a group and permission-driven prefix:

$$dist(ID, key) < 2^{n-\beta}$$



Flexible scheme for group resource management

- provides the possibility to exploit the **entire DHT space** by peers of any level for the publication of meta-data according to the original DHT principles
- a “value” associated with a certain “key” can contain either the resource itself (preferably small size data) or just some information about its location and some resource description (meta-data)
- this approach increases the number of nodes that collaborate on the maintenance of the DHT and, hence, strengthen the DHT resilience



Meta-data structure

- the DHT represents a location registry that can be used to discover where the desired resource can be found
- `uri` tag contains a routable URI with HostPort information (in order to by-pass any centralized DNS system) and other resource-related information useful for characterizing and/or accessing the resource

```
<resource>
<key>a0b1c2d3e4...</key>
<name>http://www.mynet.org/docs/file.pdf</name>
<uri>http://80.10.1.2:8080/www/docs/file.pdf</uri>
<publisher>ff0123bcae...</publisher>
<access_rights>12</access_rights>
<expires>MON Oct 06 09:00:00 GMT 2008</expires>
<updated>MON Jun 08 09:00:00 GMT 2008</updated>
<signature>fabc234678...</signature>
</resource>
```

Certificate-based DHT protection

Certificates are used to:

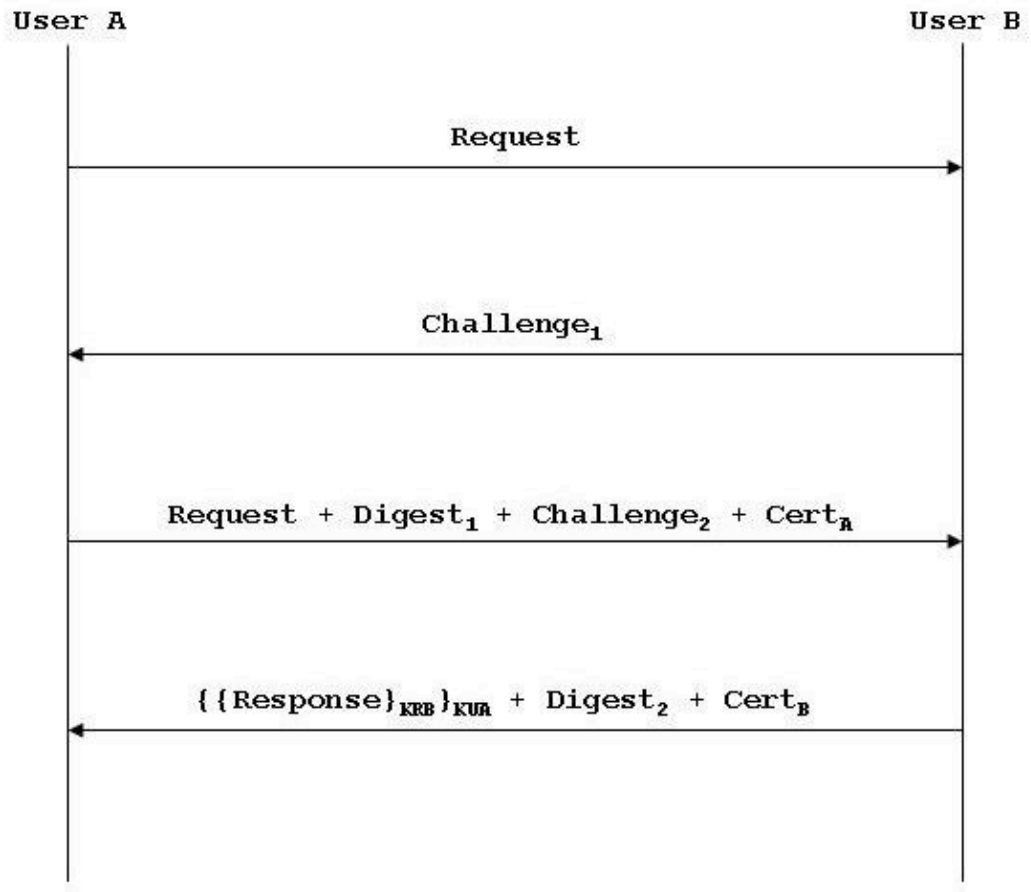
- authenticate (signing) meta-data containing information about the location, the publisher, the access rights of a resource; this allows at any time a receiving peer to verify the resource's contents through the check of the given signature
- ensure resource data protection when the actual resource is accessed after the lookup mechanism has been succeeded
- control that meta-data are available only to nodes that are granted the appropriate access privileges
- protect the DHT against poison attacks, both for resources and for routing table entries
- optionally secure all DHT operations by enforcing cryptographic confidentiality

Certificate-based DHT protection

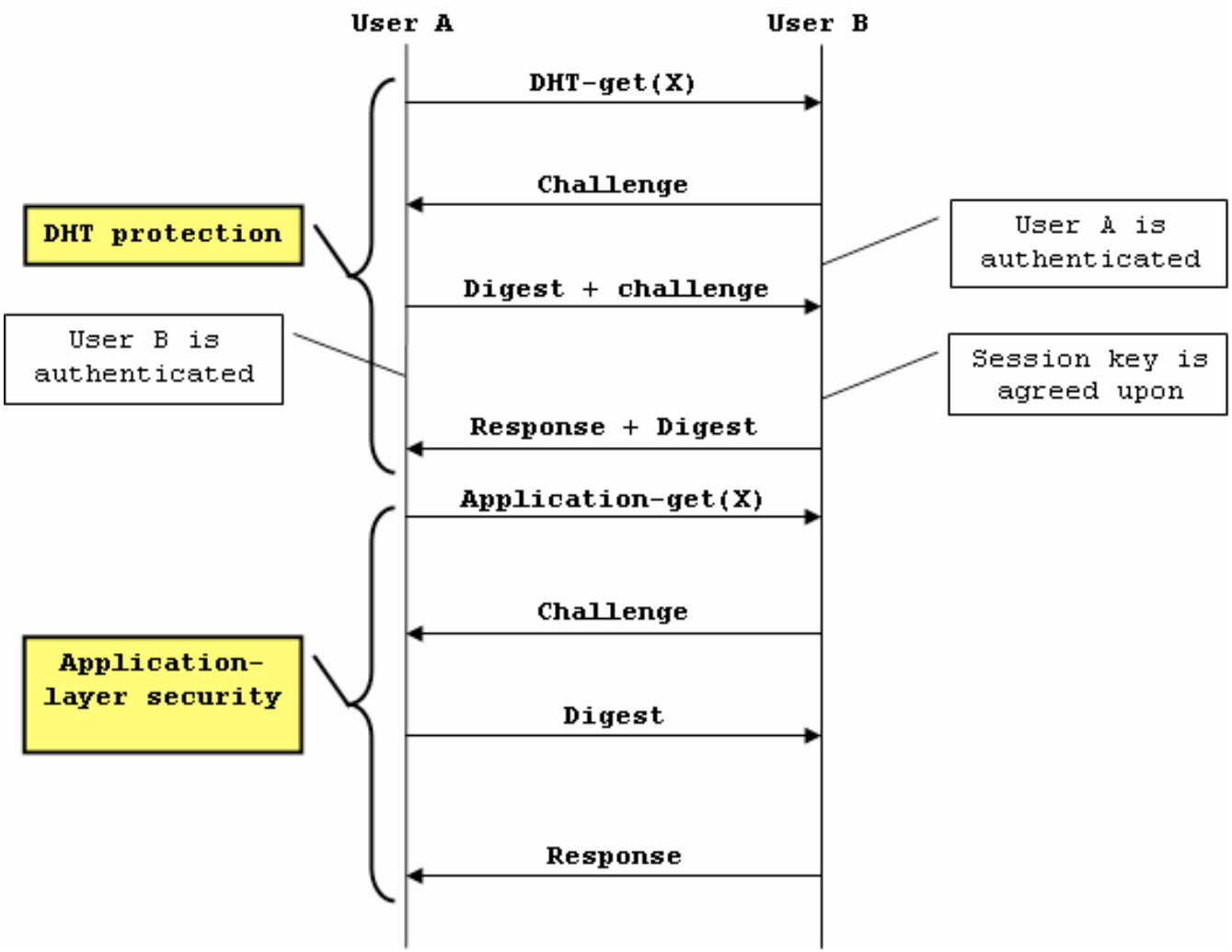
The **“join”** process is basically divided into the following steps:

- the bootstrap node verifies the certificate of the joining node and checks if it belongs to the same group;
- the bootstrap node also verifies that the node ID is not already in use by performing a PING request for the ID, thus preventing Sybil attacks
- if a joining peer contacts a bootstrap node responsible for another group, the bootstrap node, after verifying the peer’s credentials, it will redirect the request to the appropriate bootstrap node responsible for such group

Authentication mechanism inside DHT requests



Application layer security



Implementation

- the solution has been implemented in a demonstrative testbed through protocol for DHT maintenance and the protocol for resource access according to the proposed security mechanisms based on the Session Initiation Protocol (SIP)
- SIP is an extensible application-layer UDP-based signaling protocol that permits to initiate, modify and tear down multimedia sessions in a peer-to-peer fashion
- for the DHT maintenance the dSIP protocol (a specific extension of SIP) has been used in order to allow for compatibility with any DHT algorithm
- certificate issuing and authentication aspects are based on the SIP digest-authentication scheme proposed in the SIP RFC and currently being implemented and tested

Conclusions

- a new resilient resource management scheme based on Kademlia lookup mechanisms for distributed collaborative environments has been proposed
- the previous solution has been extended and generalized in order to address some common problems regarding resilience and fault tolerance of distributed collaborative systems based on DHTs
- network nodes exploit the entire DHT ID space for resource management instead of realizing storage/retrieval mechanisms only within the groups they belong to, in accordance with confidentiality principles
- to enforce DHT protection, a robust two-way authentication (and optionally confidentiality) mechanisms have been introduced