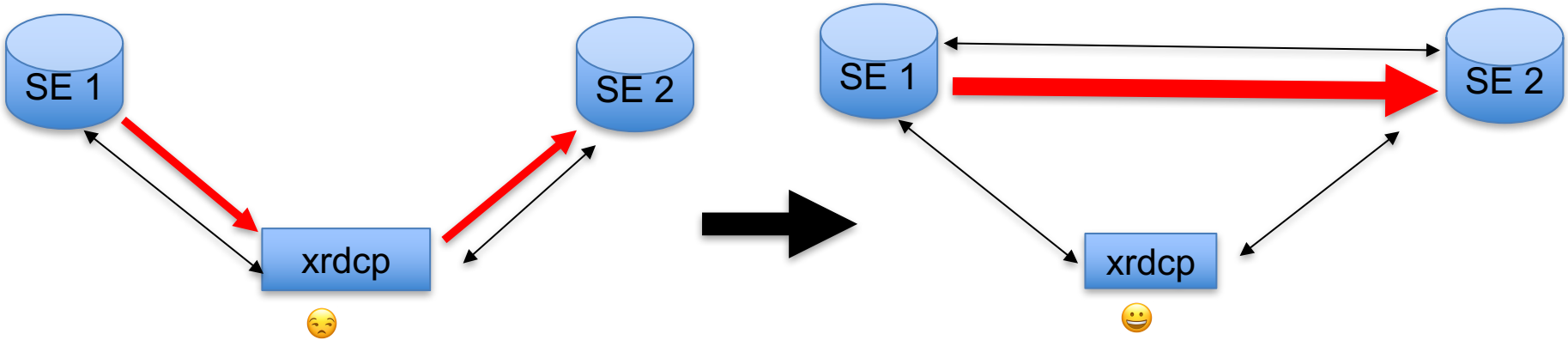
dCache.org

XrootD Third-Party Transfers Using dCache

Albert Rossi
Dmitry Litvintsev
Fermi National Accelerator Laboratory

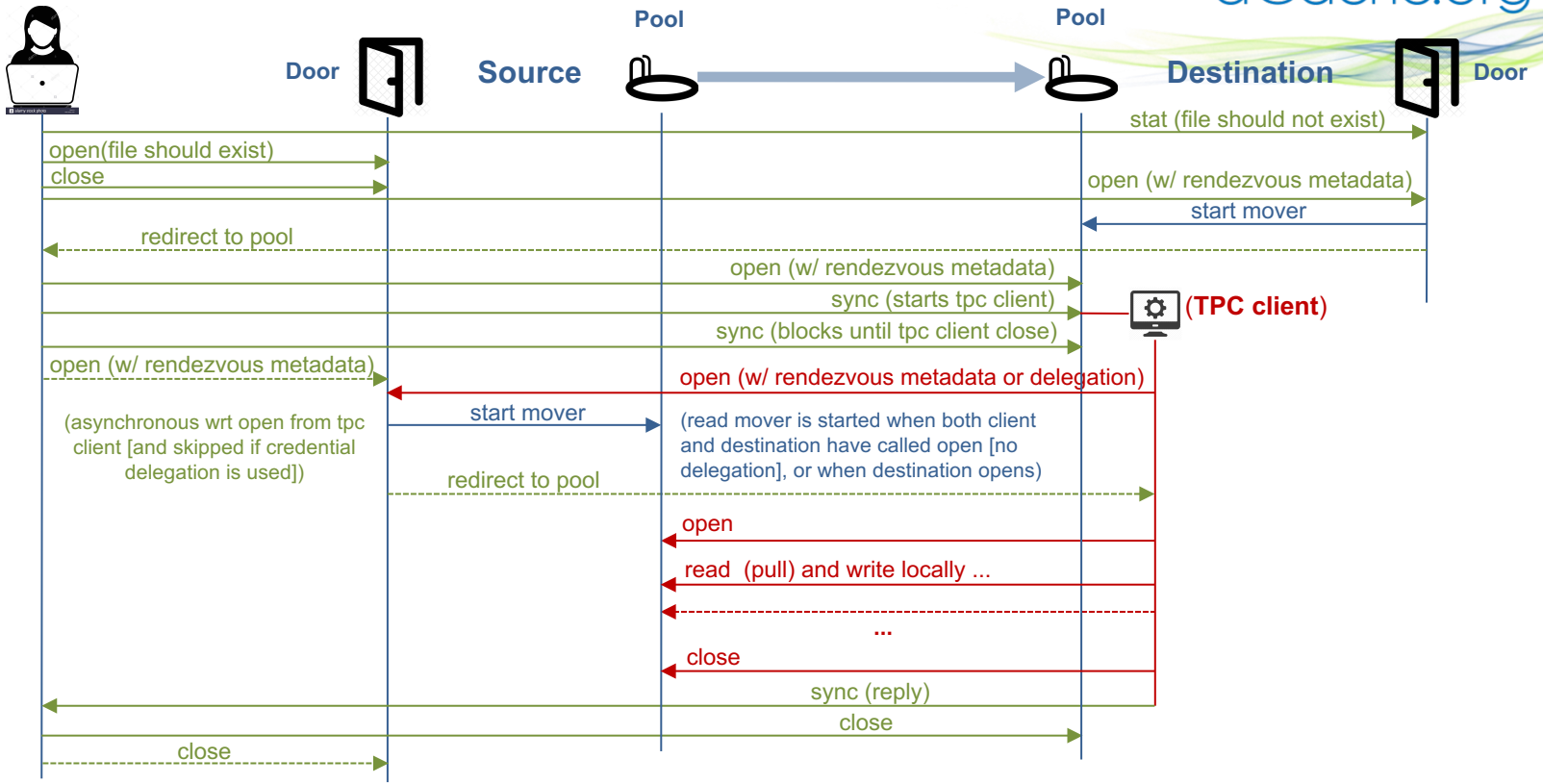
Third Party Copy (TPC). A workhorse of WAN transfers.



How it works in dCache

- User Client (e.g. xrdcp) contacts source and destination servers to see if TPC is supported
- Client generates rendezvous token, or uses delegation
- Servers communicate using the token, or delegated proxy
- Destination pulls from source using server-side embedded client

How it is implemented in dCache



TPC Authentication with dCache

dCache.org

1. Unauthenticated
2. GSI (x509 certificate-based)

dCache provides three options for (2)

a) Use proxy automatically generated from host certificate (on pools) [dCache 4.2+]
b) Use externally generated proxy (on pools) [dCache 4.2+]
c) Full proxy delegation [dCache 5.2+]

TPC Authentication with dCache

dCache.org

(1) proxy automatically generated from certificate (e.g., host)
- requires cert and key .pem on all affected pool nodes
- refreshed/regenerated by dCache
- pre-5.1: need to map all pool DNs (on all potential source endpoints!)

(2) externally generated proxy
- required on all affected pool nodes (indicated by path)
- requires external cron to refresh

(3) full proxy delegation
- no extra configuration necessary

dCache 5.2 defaults to (3) if delegation is requested/supported by the client; otherwise it will try (1) unless the path for (2) is configured.

Note that dCache 5.1+ allows for anonymous (authenticated) fallback during the server-to-server communication, so gPlazma mapping of the destination DN is not necessary in the case of 1 or 2 (it just needs to come from a recognized CA). Turn this on via property (see later slide).

Fermilab NDGF Nordic DataGrid Facility DESY Helmholtz Association neic LSDMA

Authentication with Security Level

dCache.org

XrootD security level

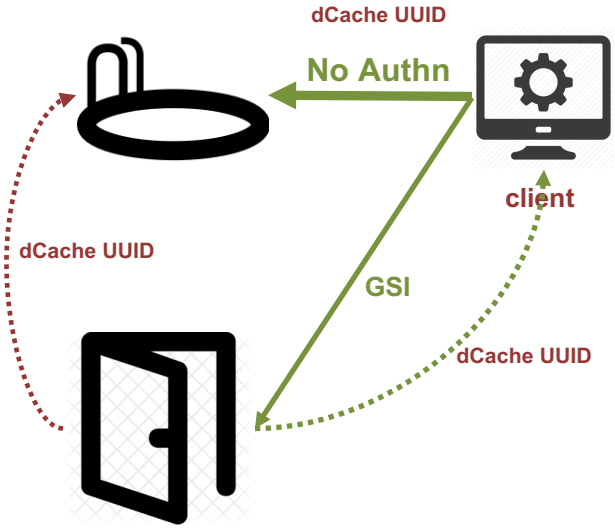
- added protection against man-in-the-middle attack
- requests preceded by signed hash which must be verified
- from minimal (level 1) to maximal (level 4), where 4 means all requests sent after the authentication handshake completes
- not just for third-party-copy

dCache 5.0+ implements security/signed hash verfication for both the door (server) as well as in the TPC client.

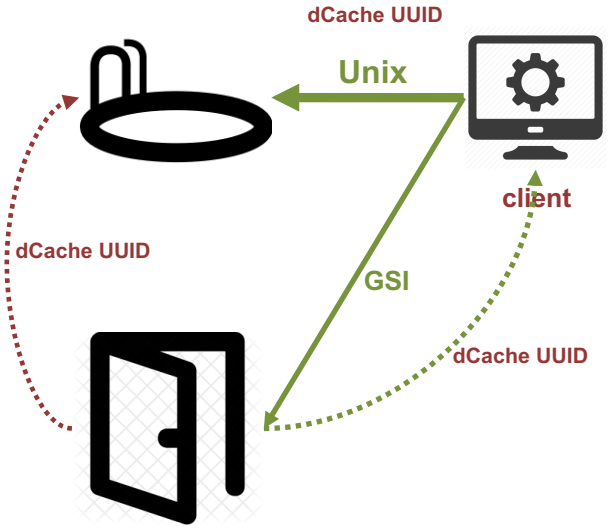
Pools: No Authn vs. Unix Authn

dCache.org

Level = 0 (off)



Level > 0 (on)



Enabling TPC (XrootD client)

dCache.org

xrdcp 4.8+

to enforce third-party copy:

```
xrdcp --tpc only root://<source> root://<destination>
```

to try third-party copy but fail over to 1-hop:

```
xrdcp --tpc first root://<source> root://<destination>
```

xrdcp 4.9+

to delegate the proxy:

```
xrdcp --tpc delegate only root://<source> root://<destination>
xrdcp --tpc delegate first root://<source> root://<destination>
```

XrootD TPC in dCache – version requirements

dCache.org

- To use dCache as the source in XrootD TPC
  update to version 4.2+ on the nodes running the xrootd doors
- To use dCache as the destination in XrootD TPC
  update to version 4.2+ on the nodes running the xrootd doors and all relevant pools
- To enable security level support
  update to version 5.0+ on the nodes running the xrootd doors and pools
- To enable anonymous authenticated reads by TPC destination client
  update to version 5.1+ on the nodes running the xrootd doors
- To enable full GSI proxy credential delegation
  update to version 5.2+ on the nodes running the xrootd doors and pools

Fermilab NDGF Nordic DataGrid Facility DESY Helmholtz Association neic LSDMA

Version Compatibility

- SLAC/XrootD servers and clients for 4.9+ are backward compatible with 4.8+, meaning they can tell (from the protocol version communicated during handshake) whether or not to (try to) use proxy delegation.

- Also true for dCache 5.2. Can be used for TPC with dCache 4.2+ and SLAC/XrootD 4.8+ (client and server), and will be able to determine whether delegation is supported or not.

- Of course, if delegation is not supported, then dCache needs to be configured to fail over correctly. This involves making a credential available on the pools, and DN recognition.

Configuring GSI for TPC -- Doors

dCache.org

- **as usual**, enable GSI on the doors via:

```
xrootd.plugins=gplazma:gsi,...
```

- **dCache as source** requires gPlazma to recognize as valid the credential of the destination (client)

- **v5.1+ –** to allow the destination client read-only access (even to read-protected files), set:

```
xrootd.plugins=gplazma:gsi,authz:none
xrootd.authz.anonymous-operations=READONLY
```

- otherwise, mapping needs to be done for all potential client endpoints when delegation is not used.

Configuring GSI for TPC -- Pools

dCache.org

- **dCache as destination** requires a pool plugin to be set:

```
pool.mover.xrootd.tpc-authn-plugins=gsi
```

- a pool-side credential must be available for the third-party client to use

option 1: proxy auto-generated from `.pem` files; these default to location for `hostcert` and `hostkey`, but can be overridden via properties beginning with:

```
xrootd.gsi.tpc.cred...
```

option 2: use externally generated and refreshed proxy; path indicated by:

```
xrootd.gsi.tpc.proxy.path
```

option 3: full proxy delegation (no further configuration necessary)

Configuring GSI for TPC -- Pools

- to support pre-4.9 `xrdcp` clients wishing to do TPC using dCache as destination, the pools must be configured using **one of the first two options; in the case of delegating clients, the pool-side credential will be ignored in favor of the delegated proxy.**

- all relevant properties for GSI are found in:

`xrootd-gsi.properties`

Configuring Security Level (5.0+)

dCache.org

- The embedded third-party client will honor signed hash verification if the source server indicates it must be observed.

- 5.0+ provides the following properties to set server/door security level:

```
dcache.xrootd.security.level={0-4}
dcache.xrootd.security.force-signing={true,false}
```

- In order to enforce a security level > 0 on the pools, the second property must be set to true

- *If one anticipates there will be TPC transfers between two dCache instances or two dCache doors*, the 'unix' plugin must be added to all the relevant pool configurations:

```
pool.mover.xrootd.tpc-authn-plugins=gsi,unix
```

See further the `xrootd.properties` file.

Fermilab NDGF Nordic DataGrid Facility DESY Helmholtz Association neic LSDMA

Troubleshooting

- The xrdcp client allows for –d 1-3 debugging output. This can be useful for some things, but unfortunately is more verbose than diagnostically revealing for things like authentication issues.
- For connection and authentication issues, check first the .access logs for both the doors and pools.
- The pinboard running at INFO level can sometimes reveal the issue: e.g.,

```
10 May 2019 10:57:03 [xrootd-net-4] [door:Xrootd-stkendca06a@xrootd-stkendca06aDomain:AAWIip4efYA] Using padded DH secret generation.
10 May 2019 10:57:04 [pool-5-thread-74] [door:Xrootd-stkendca06a@xrootd-stkendca06aDomain:AAWIip4efYA] Opening //pnfs/fnal.gov/VOs/dteam/tpctest/root/domatest/file5_6c83b12c-d4ff-4406-a144-4f9c2332393c for write
10 May 2019 10:57:12 [pool-5-thread-74] [door:Xrootd-stkendca06a@xrootd-stkendca06aDomain:AAWIip4efYA] Redirecting to /131.225.69.21:20555
10 May 2019 10:57:12 [pool-5-thread-74] [door:Xrootd-stkendca06a@xrootd-stkendca06aDomain:AAWIip4efYA] Sending the following host information to the client: 131.225.69.21
10 May 2019 10:57:13 [pool-5-thread-73] [door:Xrootd-stkendca06a@xrootd-stkendca06aDomain:AAWIip4efYA] Trying to delete //pnfs/fnal.gov/VOs/dteam/tpctest/root/domatest/file5_6c83b12c-d4ff-4406-a144-4f9c2332393c
10 May 2019 10:57:32 [Xrootd-stkendca06a-0] [door:Xrootd-stkendca06a@xrootd-stkendca06aDomain:AAWIip4efYA rw-stkendca06a-1 DoorTransferFinished 000085AEC9594B6D499EAC6E5AE988876A9B] Transfer 000085AEC9594B6D499EAC6
E5AE988876A9B@PoolName=rw-stkendca06a-1 PoolAddress=rw-stkendca06a-1@rw-stkendca06a-1Domain failed: Post-processing failed: File size mismatch (expected=1000000000, actual=0) (error code=10004)

[stkendca06a] (Xrootd-stkendca06a@xrootd-stkendca06aDomain) admin >
```

Troubleshooting

- A rather extensive output can be activated at the debug-level:

```
\s <door>,<pool>,... log set stdout org.dcache.xrootd DEBUG
```

- In addition, byte dumps of the authentication handshake can be printed by turning on trace-level debugging. This may be useful to send to the developers (me) in case of an unusual authentication failure. The byte dumps are printed to resemble those produced by the SLAC xrootd implementation.

```
\s <door>,<pool>,... log set stdout org.dcache.xrootd TRACE
```

Possible Future Work

1. Optimize door-to-door (dCache to dCache) TPC not to use the rendezvous token (similarly to the XrootD server)
2. Support SciTokens
3. Enable vector reads in the third-party-client

A special note of thanks goes to the SLAC xrootd development team for their helpful responses and collaborative spirit, in particular, Andy Hanushevsky, Wei Yang, Michal Simon and Gerri Ganis.