BitTorrent Sync: First Impressions and Digital Forensic Implications

By

Jason Farina, Mark Scanlon and Tahar Kechadi

Presented At

The Digital Forensic Research Conference

DFRWS 2014 EU Amsterdam, NL (May 7th - 9th)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment. As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

http:/dfrws.org

BitTorrent Sync: First Impressions and Digital Forensic Implications

Jason Farina*, Mark Scanlon* and M-Tahar Kechadi

jason.farina@ucdconnect.ie, mark.scanlon@ucd.ie
http://csi.ucd.ie

School of Computer Science and Informatics,
University College Dublin, Ireland.

DFRWS
DIGITAL FORENSIC RESEARCH WORKSHOP

Contribution of this Work

1. An introduction to the architecture, protocol and behaviour of the BitTorrent Sync application

2. Highlight the local files, artefacts and network traffic of interest to a digital investigator analysing a machine with the BitTorrent Sync application present including:
   - Files created during installation
   - Artefacts left behind after uninstall
   - Indicators that files have been transferred via BTSync
   - Description of the BTSync specific network traffic that may indicate live connections to remote systems or active transfers.

What is BitTorrent Sync?

- BitTorrent Sync (BTSync) is a file replication utility intended to enable remote, verifiable synchronisation of content
- Much smaller size of intended recipients than open Peer-to-Peer
- Files are not stored on any machine other than one authenticated by an authorised party
- Data synchronisation can be one or two way between participating members
- Applications available for Windows, Mac OS X, Linux, iOS, Android and Kindle Fire

Why Might BTSync be of Interest?

- Cloudless Backup Investigation
  - Optional Encryption
- Dead-Drops
  - Users post secrets online for other users to download from or upload to.
- Industrial Espionage
- Silent File Transfer
  - By turning off all optional settings and using "known hosts" a user can transfer a file over a network that blocks by address blacklist and not protocol signature.
- Encrypted P2P communication within a LAN/WAN
  - http://missiv.es/ proof-of-concept based on BTSync
- Server-less website hosting to bypass censorship
- Malicious software distribution/maintenance

Timeline of BTSync

January 2013: Closed Alpha announced. Request for testers posted on BitTorrent Blog.

April 2013: Open Alpha announced.

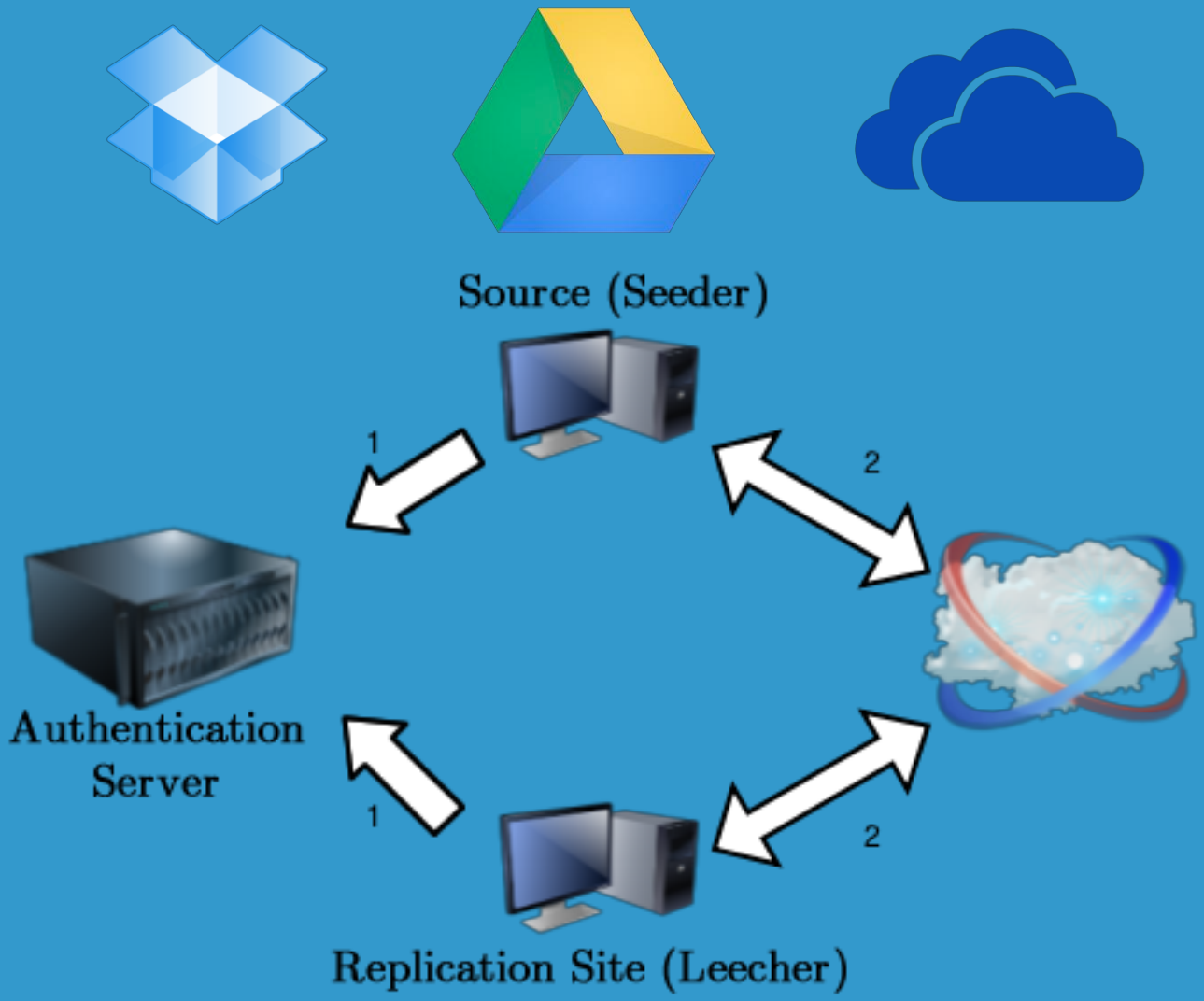
May 2013: 70 terabytes daily traffic reported. 1 PB transferred to date.

July 2013: Open Beta launched.

November 2013: 1 million users announced.

December 2013: over 2 million users, over 30 PB data synced to date.

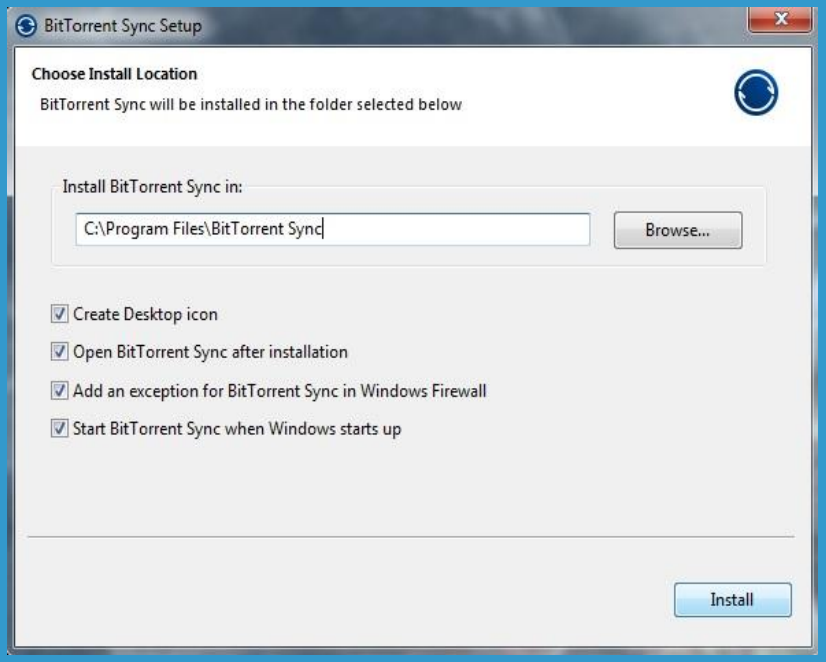
Cloud Synchronisation

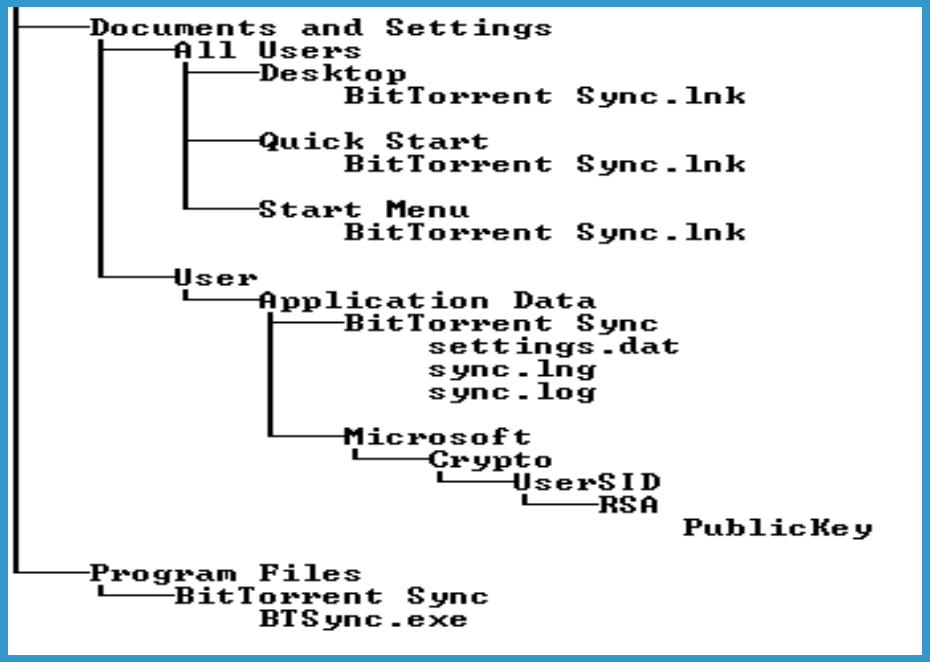


BTSync

- With BTSync each client acts as the authenticating server.
- Files are stored locally on each machine meaning storage is only limited by the hardware available to the end user.

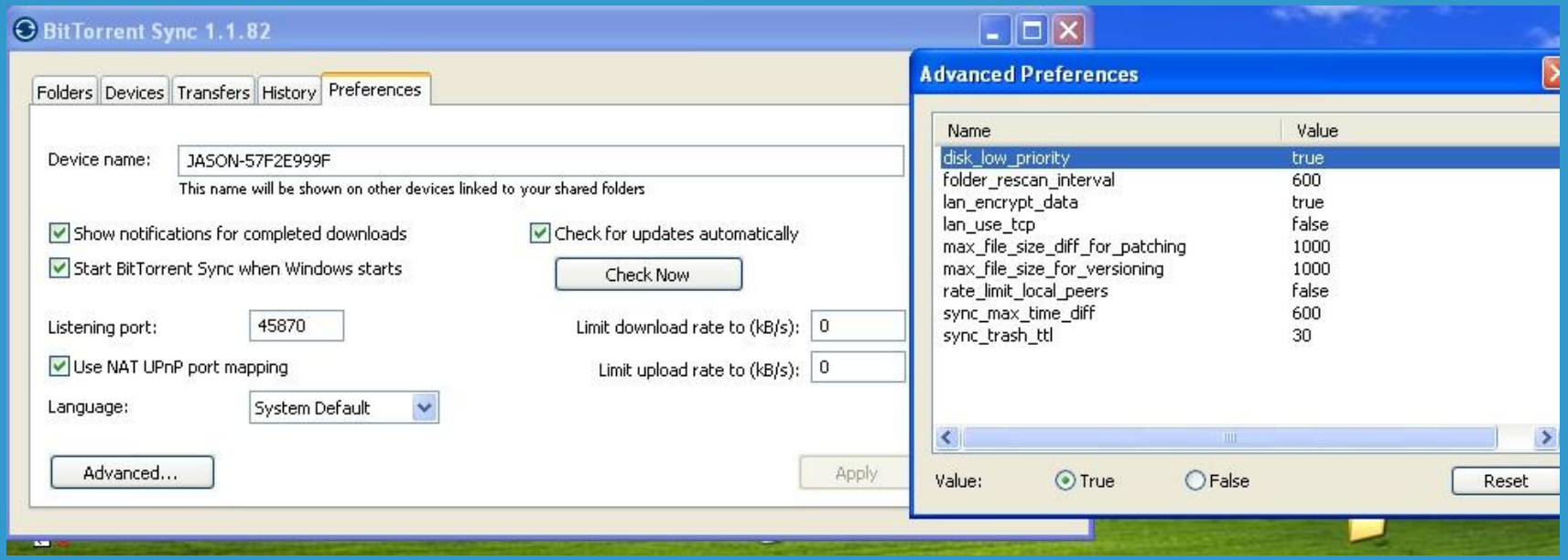
Indicators: Installation





- In Linux the **`User\Application Data\BitTorrent Sync`** folder is a hidden folder called **`.btsync`**
- If the application is running the blue circle logo will be displayed in the system tray and service **`BTSync.exe`** will be listed in Processes.

Indicators: Installation II



- This is the set of default system settings applied when installed.
- **`lan_encrypt_data`** is on by default. Only LAN traffic has the option of being unencrypted.
- **`sync_trash_ttl`** is the number of days to store deleted sync items in shares.

Bencoding

- Bencoding, or Byte Encoding, is a method of data storage used by BitTorrent in the majority of its BTSync system files.
- Each block of data consists of **`KEY:VALUE`** pairs with a numerical byte count integer before each label to indicate its length.
- Some characters have special significance if encountered outside of a **`KEY:VALUE`**. In general these would be

`d`	beginning of dictionary
`e`	end of object
`l`	begin list
`i`	start of integer terminated by `e`

- When utilised a lot of data can be easily collated in a format that is of value to both the application and an investigator.

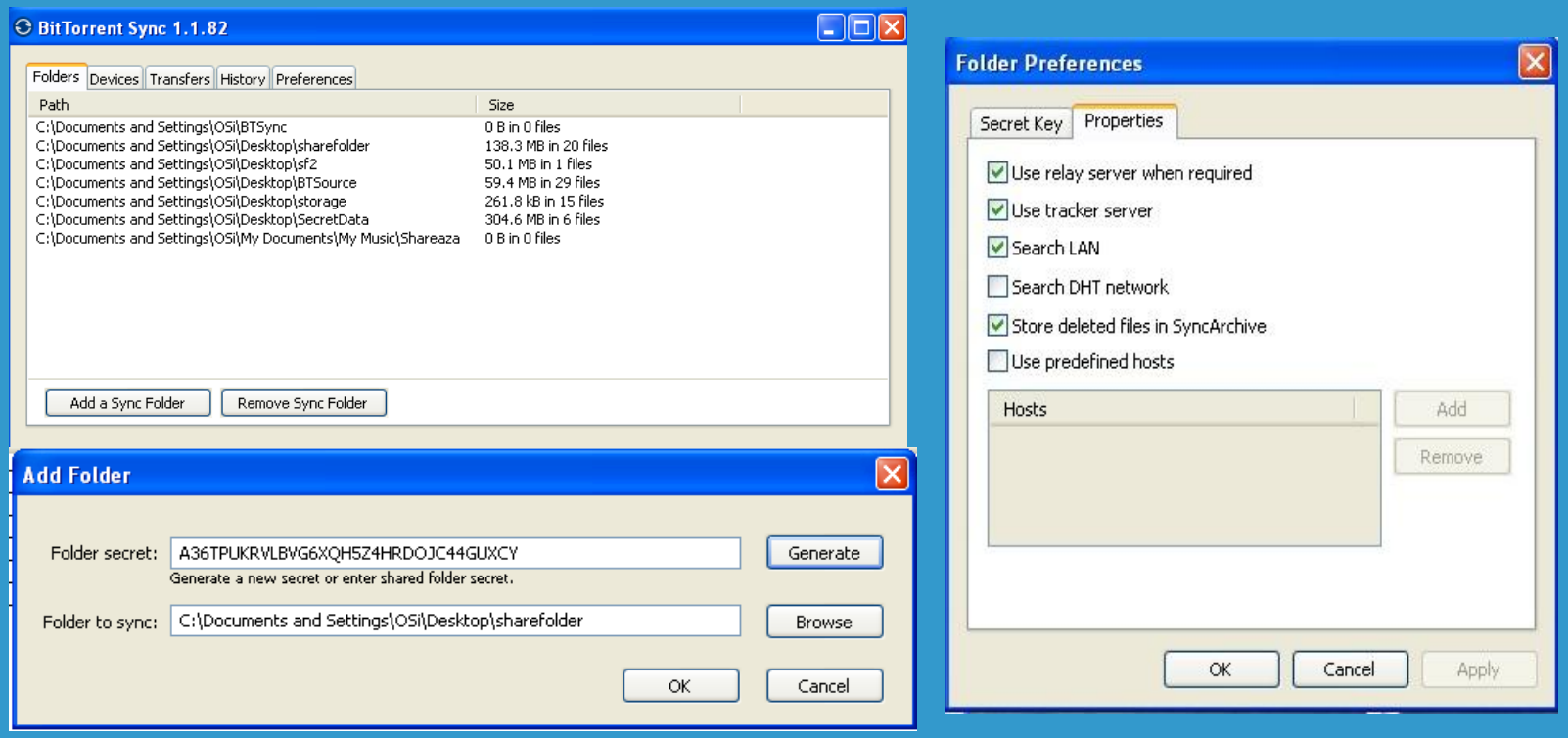
Bencoding Example

```
d10:.fileguard40:A88654AEA87C628B0D748D94457480C586AA1E5F7
:version6:1.1.826:device15:JASON-
57F2E999F7:foldersld4:path40:\\?\C:\Documents and
Settings\OSi\BTSync6:secret33:A4PMTORFNLCINYA5BP3DYKQUQFE3
TWRUU15:stopped_by_useri0e7:use_dhti0e17:use_lan_broadcast
i1e9:use_relayi1e11:use_trackeri1e15:use_known_hostsi0e11:
known_hostsle5:peersle7:invitesle11:folder_typei0e15:delet
e_to_trashi1e22:mutex_file_initializedi1ee
```

Indicators: Application Files

- Files containing information of use to an investigator are generally stored in the application folder in the User's profile.
- **`sync.pid`** - the process id **`BTSync.exe`** is currently running under
- **`settings.dat`** – A bencoded list of the current application settings including active port and port number restrictions.
- **`sync.log`** – The application log file containing times and dates of Peer discovery attempts and synchronisation details.
- **`sync.dat`** – Contains details of each share's settings including the secret associated with that share, the preferences set for that share and the **`known peers`**, if any, that have been entered. This file uses the bencoding format

Indicators: Share



- At install a default share folder is created called btsync
- Users can add a share at any time by designating any folder (and its subfolders) as a share.

Secrets

- BTSync is built around secrets and using them to control access to the data being synchronised.
- Secrets are generated by the application as a 20 byte random string that is Base32 encoded to make it human readable
- Secret Types:
  - Master Secret allows Read / Write access to a share
  - Read-only Secrets
  - A more secure option when secrets need to be transferred are 24 hour secrets that expire after 24 hours and the share can no longer be joined. Once applied they translate to normal RW or RO secrets.

```
RW:             ACHY3VFJZ3RJ3DE2CHPUGE6W7EZSRA3OR
RO:             BY6G6B7KIBGELLXE2RL65C34CAGPV7LUJ
24-hour RW:     CBJIK32CLMWF2P7JLFYRGC3JRTEZ6JLPU
24-hour RO:     CCYGZN6R67O67QB7HGLL4F5BAVA3AJ5LC
```

Custom Secrets can be generated by Base64 encoding a string or passphrase.

Indicators: Files Related to a Given Share

- Once a share has been allocated and the user has generated a secret or input one taken from elsewhere, several files are created to manage the synchronisation process:

  `.SyncID`: contains the name of the corresponding **`*.db`** file in the BTSync settings folder.

  `.syncarchive` (folder): files that are deleted on a source system are moved into this folder when the state is synchronised to the local machine.

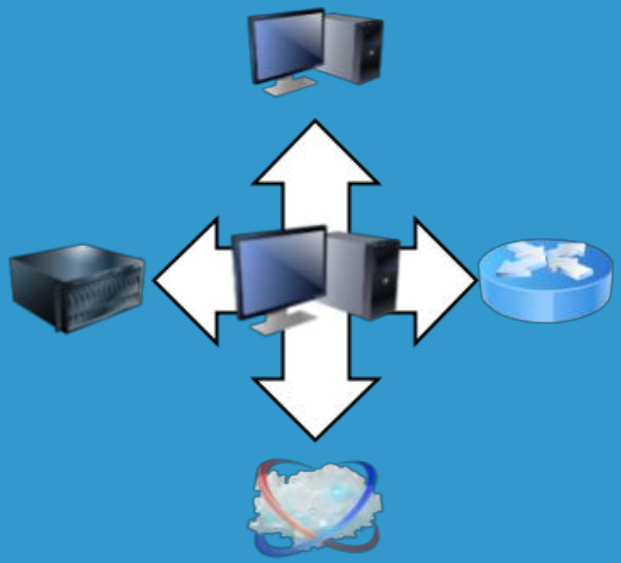
  `.SyncIgnore`: designate a file or folder to omit when synchronising downstream.

Indicators: Share – Application links

- Once a share has been created the contents of the share are indexed and stored in a SQLite3 database in the BTSync folder in the user profile.
- The name of the database is a 40 character string derived from the secret.
- This `[ID].db` file contains a bencoded listing of each file in the share and the metadata needed to process and manage synchronisation
  - Invalidated
  - Deletion status
  - Path
  - System permissions
  - Timestamps for synchronisation
  - 20 byte hash for torrent transmission
  - 64 byte file signature

Peer Discovery

- BTSync uses the same methods of peer discovery as BitTorrent.



Tracker : if the tracker option is left enabled the peer will register its share with a tracker while requesting a list of peers with the same share available. The tracker is contacted by connecting to **`t.usyncapp.com`** which at the time of writing resolved to two servers hosted in Amazon's EC2 cloud.

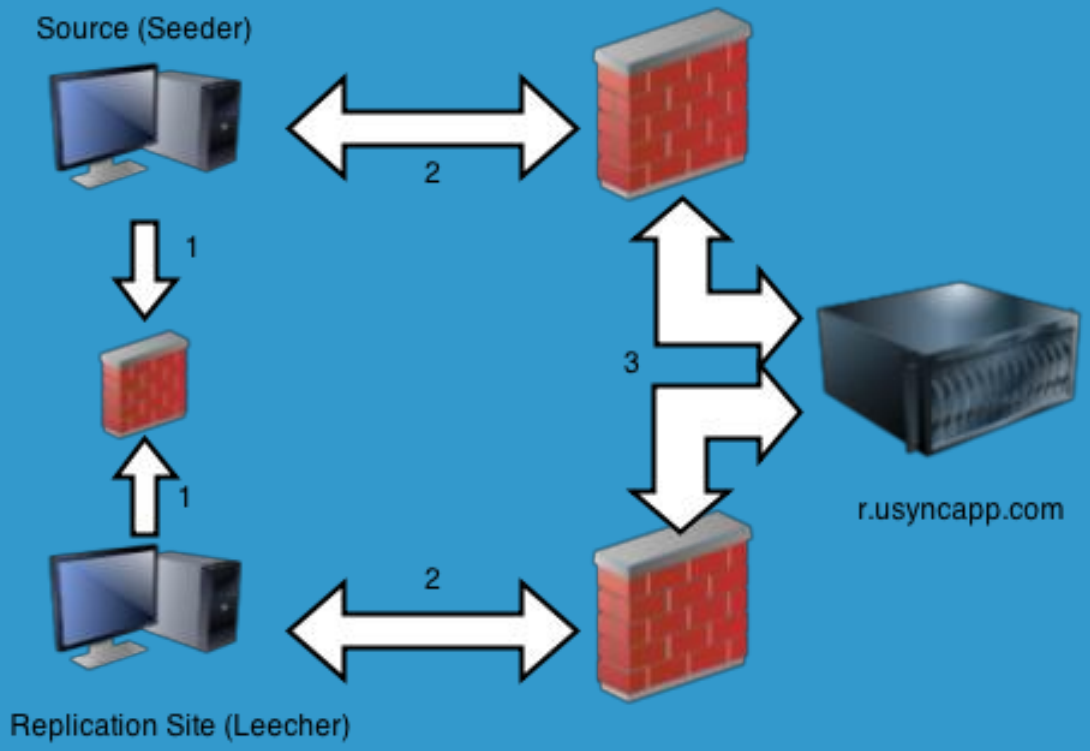
LAN Discovery: If the **`Search LAN`** option is left enabled, BTSync will advertise its shares on the LAN by sending a packet to the multicast address: `239.192.0.0` on port 3838. Any BTSync installations with a matching share ID will respond and identify themselves.

DHT : The Third method of discovery is to use DHT. With this method BTSync will register itself with any peer it comes in contact with the string **`SHA-1(Secret):IP:Port`** .

Known Peers: The Last and most secure method Peer Discovery is the option to **`use known peers`**. When this option is enabled (not default), BTSync will communicate directly with the **`IP:Port`** entries listed.

Connection

- Once a peer has been discovered BTSync attempts to establish a connection. If a direct connection is not possible because of a firewall or some NAT configuration, BTSync offers the use of a relay located at the URL **`r.usyncapp.com`**



Don’t corporate firewalls already stop this if they block torrents?

- Perimeter firewalls are usually only configured to block/inspect traffic outside the organisation.
- BTSync traffic is encrypted after the initial key exchange meaning the actual torrent stream is never available for the firewall to inspect or analyse. Without draconian firewall rules, BTSync traffic may only show as bandwidth usage.
- BTSync can be used to transfer data internally, encrypted in transit, and optionally encrypted on the receiving end.
  - This would allow data to be transferred to a weaker / more open area of a network where it could be transferred to another media (USB lockdown, Guest LAN, etc.).

BTSync vs. Regular BitTorrent

	BitTorrent Sync	Regular BitTorrent
Peer Discovery	PEX, DHT, Tracker, Known Peers	PEX, DHT, Tracker, Known Peers
Entry Points	Shared Secret (QR Codes, SMS, email, IM, website, offline, etc.)	`*.torrent` metadata file/magnet URI (indexing website)
Access Rights Control	Private/Public RW/RO/RO Encrypted	Private/Public RO
Transfer Encryption	Mandatory WAN, Optional LAN	Not Native
Firewall Bypass	Relay Server, Encrypted Transfer, Known Peers	Not Native

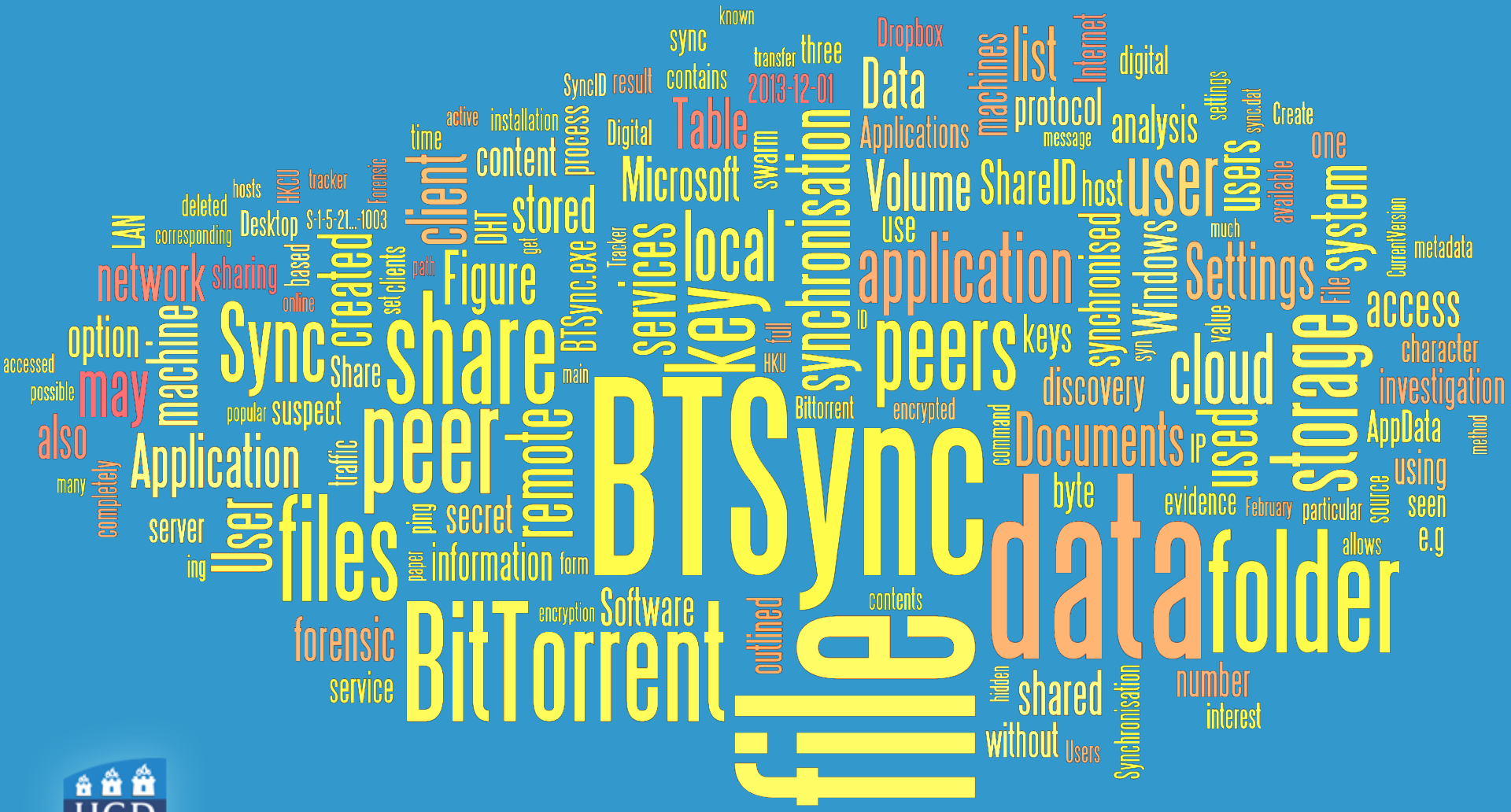
Deletion (remnants)

- BTSync does provide an uninstaller.
- Methods of uninstallation:
  - Delete share files:
    - Local deletion: the details of the file, including the date it became invalid, will be stored in the share database file.
    - Remote deletion: the local file will be moved to the `.syncarchive` folder and stored for the default 30 days. Any file metadata will still be stored in the file database
  - Share Removed: The share management files will still be stored in the folder.
  - Folder deleted: the log files will list the last time the files were synchronised, the database files will store the folder content metadata.
  - Application deleted: Settings may still be stored in the settings folder. Share management files will still be present in share folders.

Future Work

- BTSync Network Investigation Methodology
- Investigation Utility
- Automated Share Detection through Network Packet Sniffing/Deep Packet Inspection
- Crawling of active nodes involved in the same share
  - Enumeration and Geolocation
- Extending the evidence recovery window through remote evidence recovery

Thank you



jason.farina@ucdconnect.ie, mark.scanlon@ucd.ie