

FileSender

Browser based large file sharing

The Commons Conservancy

Dr Ben Martin

[Upload](#)[Guests](#)[My Transfers](#)[Admin](#)[Help](#)[About](#)[Log-off](#)

4.8 GB out of 23.3 GB used, 18.5 GB remaining

drag & drop your files here

[Clear all](#)[Select files](#)

From : ben@localhost.localdomain

File Encryption (beta)

Password :

[Generate password](#)

Show / Hide Password

File Encryption is end to end. Your files are encrypted in your web browser. It is up to you to send the encryption password to the recipient(s) as we do not store any passwords.

File Encryption will significantly impact performance of your browser and/or device for the sender and receiver(s).

Encrypted Files equal to or greater than 4GB may not be downloadable due to the limitations of the web browser.

Expiry date:

Notify me when expired

Notify me when upload is done

Notify me upon downloads

Send me a report when expired

Get a link instead of sending to recipients

[Advanced settings](#)

Send me daily statistics



Send



1.8 GB out of 23.3 GB used, 21.5 GB remaining





> Currently available transfers

+	Transfer ID	Recipients	Size	Files	Downloads	Expires	Actions
+	36039	Anonymous	30 B	df	0	14/04/2018	
+	36037	Anonymous	30 B	df	0	14/04/2018	
+	36024	Anonymous	30 B	df	0	13/04/2018	
+	36023	Anonymous	30 B	df	0	13/04/2018	
+	36022	Anonymous	30 B	df	0	13/04/2018	
+	36021	Anonymous	30 B	df	0	13/04/2018	
+	36020	Anonymous	90 B	df2 df3 df1	0	05/04/2018	
+	36019	Anonymous	1.3 GB	Fedora-Live-Workstation-x...	1 (See all)	21/03/2018	
+	35982	Anonymous	7.1 kB	ddd	0	14/03/2018	

[More...](#)

> Closed transfers

+	Transfer ID	Recipients	Size	Files	Downloads	Expires	Actions
+	36018	Anonymous	1.3 GB	Fedora-Live-Workstation-x...	1 (See all)	12/03/2018	
+	36017	Anonymous	1.3 GB	Fedora-Live-Workstation-x...	1 (See all)	12/03/2018	

+ Transfer ID	Recipients	Size	Files	Downloads	Expires	Actions
<div style="text-align: right;">    </div> <p> - Transfer ID : 36039 Created : 05/04/2018 Expires : 14/04/2018 Size : 30 B Sender email : testdriver@localhost.localdomain Options : <ul style="list-style-type: none"> • Notify me when expired • Notify me when upload is done • Notify me upon downloads • Send me a report when expired • Allow recipients to receive download complete emails • Get a link instead of sending to recipients Download link : <div style="border: 1px solid gray; padding: 2px; display: inline-block; margin-top: 5px;"> https://192.168.0.200/filesender/?s=download&token=66b6052f-a4e7-d731-f15c-0386b4 </div> </p> <p>Transfer audit</p> <div style="border: 1px solid gray; border-radius: 5px; padding: 5px; display: inline-block; margin-top: 10px;">  See the transfer logs </div>						

Transfer audit ✕

Date	Action that happened	IP address
05/04/2018 08:26:46	Transfer was created	192.168.0.200
05/04/2018 08:27:03	File df (30 B) uploaded (took 0s)	192.168.0.200
05/04/2018 08:27:03	Upload ended	192.168.0.200
05/04/2018 08:27:03	Transfer became available (took 17s)	192.168.0.200

 Send to my email

Close

Why

- User only needs Web browser
- Full control of content
- Upload resume
- Notification on upload, download, etc
- Share with explicit groups
- Browser-to-Browser encryption of data AES-256
- SAML for auth scale
- Allow guests to upload to your server
- GDPR by default, auto generated “about privacy” page

How

- Server side is PHP
- Client side JS with light widgets
- Database is Maria 10.0+ or PostgreSQL \$recent
- Web server is Apache or NGINX
- Auth is SAML or hacked^TM

Chunks

- Minimal range of bytes to commit to server
- Size of memory needed in browser (blob.`slice`)"
- Somewhat hidden from users

Uploading

- TeraSender - Web Workers
 - Number of chunks upload at once
- PUT chunks directly
- PUT wholefile
- REST client

Server Storage

- Contiguous file
- Chunked (5mb) files
 - Above either raw or in nesting path for NFS dispatch
- External script
- Ceph (aarnet)
- Cloud – Azure, S3

Downloading

- One or more files listed per transfer
- On the fly zip64 archive creation
- Links for console download if desired

Dragons

- Robust, auto resume and fast upload cross browser is HARD
- eg, Disable wifi during upload
- Mixed browsers and racing
- Long uploads can exceed auth session times
- Web crypto support W3C

Database Design

userpreferences

|

|

|

v

guests

<----- transfer -> files

|

v

recipients

shredfiles,

auditlogs (subj, pred, action),

trackingevents (pred, action)

statlogs, new aggregatestats

translatableemails (class, id => data)

Database fun

- Moved to synth id for userpreferenceses pk
- Moved saml authid to another table
- Added RI about the place
- Added secondary indexes
- Added views to ease query maintenance
-
- TODO
- Userpreferences/guest cleanup
- PHP ↔ SQL worlds

ORM

```
protected static $dataMap = array(
    'id' => array(
        'type' => 'uint',
        'size' => 'medium',
        'primary' => true,
        'autoinc' => true    ),
    'email' => array(
        'type' => 'string',
        'size' => 255    ), ...
```

ORM

All(), delete(),
FromDB(), toDB()
save()

client class override __get() and 'reach out'
db migration based on metadata

DBI Smoothing

```
public static function datediff( $f1, $f2 ) {  
    if(self::isPostgress()) {  
        return "extract(day from " . $f1 . "-" . $f2 . " )";  
    }  
    if(self::isMySQL()) {  
        return "DATEDIFF(" . $f1 . "," . $f2 . ")";  
    }  
    throw new DBIBackendExplicitHandlerUnimplementedException(  
        'SQLUNIMP datediff() called on unsupported backend');  
}
```

Uploading

- TeraSender overview

TeraSender

- Create a number of Web Workers
- Logically files split into 'chunks' each of which is given to a worker
- Each chunk is committed server side on success so we can resume the whole upload* at any time
- Workers can work on a single file at once or across multiple files
- Encryption is done before upload

TeraSender upload

- `upload_page.js` `onClick()` → `ui.startUpload()`
 - `Transfer.start()`
- One of these depending on config & browser
 - `terasender.start()`
 - `uploadByChunks()`
 - `uploadWholeFile()`

Teraender.start(transfer)

- Start()
 - createWorker() many times
- createWorker()
 - w = new Worker()
 - w.OnMessage();
 - w.sendCommand('start')

Worker.start

- OnMessage (start) → requestJob()
- RequestJob() → sendCommand('requestJob');

Ts.giveJob

```
var job = this.allocateJob(worker);

if( job ) {
    this.sendCommand(workerinterface, 'executeJob', job);
} else {
    workerinterface.status = 'done';
    workerinterface.terminate();
}
```

ts.allocateJob

- If worker already has a file keep it on that file
- Otherwise find a file with data to upload
- The 'job' is the next 'chunk' of the file to upload

Worker.executeJob

Data = Slice the file.blob to get the chunk

Xhr = createXhr()

Data = Encrypt Data if desired

Xhr.send(Data)

xhr.onreadystatechange()

→ ok, auth token, error (sendmsg), restart?, 200 → s(jobExecuted)

xhr.onprogress() → sendmsg(progress)

ts.jobExecuted

- Call `evalProgress()` to update UI progress data
- Fall through to `requestJob()` in case statement

Security

- Mixed auth model
- Email a group or “get a link”
- E2E crypto

Crypto

- Currently each blob crypted with aes-256-cbc using a random iv
→ GCM + AEAD

```
crypto_app().encryptBlob( array, password, cb )
```

Encrypt (current)

```
generateKey(password, function (key, iv) {  
  crypto.subtle.encrypt({name: $this.crypto_crypt_name, iv: iv}, key,  
value).then(  
  function (result) {  
    var joinedData = ....crypto_common()  
      .joinIvAndData(iv, new Uint8Array(result));  
  
    var btoaData = btoa(abtoString(joinedData));  
    callback(btoaData);  
  }  
});
```

Password → { key, iv }

```
generateKey: function (password, callback) {  
  var iv = getRandomValues(new Uint8Array(16));  
  crypto.subtle.digest({name: this.crypto_hash_name},  
                        stobv(password)).then(function (h) {  
    crypto.subtle.importKey("raw", h,  
                            {name: $this.crypto_crypt_name, iv: iv},  
                            false, ["encrypt", "decrypt"]).then(function (key) {  
                      callback(key, iv);  
                    }  
  }  
}
```

Genergate Key 2018

```
crypto.subtle.importKey( 'raw', UI.get( password),  
    {name: 'PBKDF2'}, false, ['deriveBits', 'deriveKey']).then(function(dkey) {
```

```
    crypto.subtle.deriveKey(  
        { "name": 'PBKDF2', "hash": 'SHA-256',"salt" saltBuffer },  
        dkey,  
        { "name": 'AES-CBC', "length": 256, iv: iv },  
        false, // key is not extractable  
        [ "encrypt", "decrypt" ] // features desired  
    ).then(function (key) {  
        callback(key, iv);
```

Decryption

```
generateKey(password, function (key) { ...  
var value = ...separatelyFromData(encryptedData[i]);  
crypto.subtle.decrypt(  
  {name: $this.crypto_crypt_name,  
    iv: value.iv},  
  key,  
  value.data).then( ... );
```


Encryption future

- Current: PBKDF2 (Password-Based Key Derivation Function 2)
- Future: password AND
 - doc id
 - Transfer id
 - sender email
 - other immutable data
- User passphrase: Selectable hashing level
- Using generated key from entropy with user readable display format (recommended)

Encryption GCM (future)

Key + IV **must** always be unique
counter **must** never wrap
hardware support for GCM

128 bit IV = 96 bits random() and 32 counter.
5mb chunks gives 20tb files

GCM operation

```
key = hased_password_or_secure_generated();
```

```
rand96 = randbits( 96 );
```

```
counter = 0
```

```
iv = combine( { rand96, counter } )
```

```
otp = Encr( key, iv, plaintext=counter )
```

```
ct = pt XOR otp
```

```
counter++
```

I18ns

language/en_au/lang.php

```
$lang['about'] = 'About';
```

```
$lang['about_page'] = 'About';
```

...

language/en_au/guest_cancelled.mail.php

...

A voucher from {guest.user_email} has been cancelled.

I18ns

import-translation-for-language.php en_au poeditor.export.php

export-all-terms.php terms.txt

convert-one-per-line-terms-to-json.php terms.txt terms.json

send terms.json to poeditor

Future Directions

- UI refresh
- Maybe mobile app, “send to” file sender
- Integration to endpoints (auto youtube etc)
- E2E encryption from CBC to CGM (NIST SP 800-38D)
- Docker images for easy setup and go. Migration docs for database upgrades in existing docker images
- More SAML info
- Maybe run the whole thing in Cloud as possibility
- Session clone?