Protecting “Cloud” Secrets With Grendel

http://github.com/wesabe/grendel
Sam Quigley
Square
@emerose

Coda Hale
Yammer, Inc.
@coda
You store private information.
<table>
<thead>
<tr>
<th>User</th>
<th>Password</th>
<th>Account</th>
<th>Score</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ryo</td>
<td>roc***</td>
<td><a href="mailto:ryo@rockyou.com">ryo@rockyou.com</a></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Jia</td>
<td>DuF***</td>
<td><a href="mailto:jia@rockyou.com">jia@rockyou.com</a></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Kazu</td>
<td>nlj***</td>
<td><a href="mailto:kazu@rockyou.com">kazu@rockyou.com</a></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Lance</td>
<td>frU***</td>
<td><a href="mailto:lance@rockyou.com">lance@rockyou.com</a></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Raymond</td>
<td>h45***</td>
<td><a href="mailto:raymond@rockyou.com">raymond@rockyou.com</a></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>James</td>
<td>te9***</td>
<td><a href="mailto:james@rockyou.com">james@rockyou.com</a></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Ernest</td>
<td>7R0***</td>
<td><a href="mailto:ernest@rockyou.com">ernest@rockyou.com</a></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UserAccount</th>
<th>Email</th>
<th>Password</th>
<th>Account</th>
<th>Score</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:jennaplanerunner@hotmail.com">jennaplanerunner@hotmail.com</a></td>
<td>mek*****</td>
<td>myspace</td>
<td>0</td>
<td>bebo.com</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:phdlance@gmail.com">phdlance@gmail.com</a></td>
<td>mek*****</td>
<td>myspace</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:jennaplanerunner@gmail.com">jennaplanerunner@gmail.com</a></td>
<td>mek*****</td>
<td>myspace</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:teamsmackage@gmail.com">teamsmackage@gmail.com</a></td>
<td>pro*****</td>
<td>myspace</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ayul@email.com">ayul@email.com</a></td>
<td>kha*****</td>
<td>myspace</td>
<td>1</td>
<td>tagged.com</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:guera_n_negro@yahoo.com">guera_n_negro@yahoo.com</a></td>
<td>emi*****</td>
<td>myspace</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:beyootifulgirl@aol.com">beyootifulgirl@aol.com</a></td>
<td>holi***</td>
<td>myspace</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:dani@gmail.com">dani@gmail.com</a></td>
<td>wso*****</td>
<td>wikipedia</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User</td>
<td>mediaman1961</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:mediaman@4sol.com">mediaman@4sol.com</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phone</td>
<td>1599</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Mark Bruno</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Street</td>
<td>21st Ave Suite 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>City</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>AZ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zip</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td>United States</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedback</td>
<td>100 (99.1%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registration site</td>
<td>US</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User status</td>
<td>Confirmed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power seller</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payment method</td>
<td>CreditCard</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Credit Card CVV2</td>
<td>2752 03/2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Id verified</td>
<td>false</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Store</td>
<td>false</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registered on</td>
<td>2005-05-26T16:29:49.008Z</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paypal</td>
<td>Verified ()</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SGI Inc. - enoconI gnitraeneG rof snoltulo5

credit card numbers
BANK LOGIN USA AND EU WITH UK ALL HERE AVAILABLE NOW

BALANCE IN CHASE ......................... 7000$ TO 15500$ ========= 225$
BALANCE IN WASHOVIA .................... 2400$ TO 8000 =========== 125$
BALANCE IN BOA .......................... .7500$ TO 25000$ ========= 400$
BALANCE IN CREDIT UNION ............... ANY AMOUNT ========= 300$
BALANCE IN HALIFAX ...................... ANY AMOUNT ========= 300$
BALANCE IN COMPASS ..................... ANY AMOUNT ========= 400$
BALANCE IN WELSFARGO ................... ANY AMOUNT ========= 400$

YOU CAN CONTACT FOR MANY MORE OTHER BANK LOG YOU NEED... 1 COMERSUS SOFTWARE WITH BANK LOG IN AND BANK CREDIT CARD CODE ========= 2000$
2 COMERSUS SOFTWARE WITHOUT BANK LOG IN AND BANK CREDIT CARD CODE ========= 1500$
3 NEW WESTERN UNION HACKING BUG FOR WORLD WIDE TRANSFER ========= 500$
4 NEW PAYPAL LOG IN HACKWARE FOR HACKING FRESH PAYPAL ========= 650$
5 NEW SHOP ADMIN HACKWARE FOR HACKING ONLINE SHOP FOR CREDIT CARD ========= 2000$
6 NEW CREDIT CARD AMOUNT CHECKER FOR PEOPLE WANTING TO KNOW AMOUNT ON CC ========= 320$
7 NEW CREDIT CARD VALIDATOR FOR VALIDATING ANY FULL CC INFO ========= 320$

OTHER SERVICE
INCLUDE .......................................................... .

BANK TO BANK TRANSFER TO ANY USA BANK
BANK TO BANK TRANSFER TO ANY UK BANK
BANK TO BANK TRANSFER TO ANY EURO COUNTRY BANK

bank account info
social security numbers
Aarskog Syndrome - More info
Aase Syndrome - More info
Abdominal Aortic Aneurysm (AAA)
Abdominal Bloating
Abdominal Cramps - More info
Abdominal Hernia
Abdominal Mass
Abdominal Pain
Abdominal Swelling
Abducens Nerve Palsy - More info
Abetalipoproteinemia - More info
Abnormal EKG

Enter a condition or symptom:
Example: diabetes

Select from the list:

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Medications</th>
<th>Allergies</th>
<th>Procedures</th>
<th>Test results</th>
<th>Immunizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aarskog Syndrome</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aase Syndrome</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abdominal Aortic Aneurysm (AAA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abdominal Bloating</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abdominal Cramps</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abdominal Hernia</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abdominal Mass</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abdominal Pain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abdominal Swelling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abducens Nerve Palsy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abetalipoproteinemia</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abnormal EKG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

health care information
Management Meeting Agenda (6.9.2009)

- Review Action Items from Last Week (10 Minutes)
- End of Year Goals
- ToS, API: review first decisions (CRS, JC)
- Moderation platform next steps (CRS, GP)
- Hosebird: engage news companies (MSNBC, AP) to prove breaking news power, drive exposure? (CRS)
- Promotional give/get (evolving the definitions box) (CRS)
- Tweet vs Twitter + structured trademark guidelines (CRS, Biz)

Google Conversation Update (from last week):

- Two hour meeting last week w/ [REDACTED]
- Agreed: An agreement for some period of time makes sense - with our parameters
- Possibly charge for data
- Search results page needs to be great - better than the landing pages on Google

confidential business documents

but also...
personal conversations

Yeah so if you ever need info about anyone at Harvard
Just ask.
I have over 4,000 emails, pictures, addresses, SNS

What? How'd you manage that one?

People just submitted it.
I don't know why.
They "trust me"
Dumb [redacted]
videos

<table>
<thead>
<tr>
<th>Item</th>
<th>Code</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trojan</td>
<td>002260093250H</td>
<td>5.58</td>
<td></td>
</tr>
<tr>
<td>Mouth Guard</td>
<td>007236301241</td>
<td>0.97</td>
<td></td>
</tr>
<tr>
<td>Doe Urine</td>
<td>004944393684</td>
<td>4.84</td>
<td></td>
</tr>
<tr>
<td>Keychain Cam</td>
<td>087878400019</td>
<td>13.00</td>
<td></td>
</tr>
</tbody>
</table>

**Subtotal** 24.39

**Tax 1 7.800 %** 1.90

**Total** 26.29

VISA TEND 26.29

purchase histories
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Found site</th>
</tr>
</thead>
<tbody>
<tr>
<td>marriage counseling</td>
<td>family-marriage-counseling.com</td>
</tr>
<tr>
<td>spy on the wife</td>
<td><a href="http://www.chatcheaters.com">www.chatcheaters.com</a></td>
</tr>
<tr>
<td>spy recorders</td>
<td><a href="http://www.spy">www.spy</a> supplystore.com</td>
</tr>
<tr>
<td>postal service stamps</td>
<td><a href="http://www.usps.com">www.usps.com</a></td>
</tr>
<tr>
<td>tracking cell phone numbers</td>
<td><a href="http://www.infowars.com">www.infowars.com</a></td>
</tr>
<tr>
<td>google</td>
<td><a href="http://www.google.com">www.google.com</a></td>
</tr>
<tr>
<td>dealing with cheating wives</td>
<td><a href="http://www.webmd.com">www.webmd.com</a></td>
</tr>
<tr>
<td>cheating wives</td>
<td>dir.yahoo.com</td>
</tr>
<tr>
<td>marriage counseling</td>
<td>answers.google.com</td>
</tr>
<tr>
<td>fitness gyms</td>
<td>dmoz.org</td>
</tr>
<tr>
<td>tampa bay fitness clubs</td>
<td>tampabay.citysearch.com</td>
</tr>
<tr>
<td>saving a marriage</td>
<td><a href="http://www.divorcebusting.com">www.divorcebusting.com</a></td>
</tr>
<tr>
<td>sexual techniques</td>
<td><a href="http://www.angelfire.com">www.angelfire.com</a></td>
</tr>
<tr>
<td>texas</td>
<td><a href="http://www.traveltex.com">www.traveltex.com</a></td>
</tr>
<tr>
<td>new bedford newspapers</td>
<td><a href="http://www.mondotimes.com">www.mondotimes.com</a></td>
</tr>
<tr>
<td><a href="http://www.barnes">www.barnes</a></td>
<td><a href="http://www.barnesandnoble.com">www.barnesandnoble.com</a></td>
</tr>
<tr>
<td>stop your divorce</td>
<td><a href="http://www.stopyourdivorce.com">www.stopyourdivorce.com</a></td>
</tr>
<tr>
<td>alchohol withdrawl</td>
<td><a href="http://www.carboniq.com">www.carboniq.com</a></td>
</tr>
<tr>
<td>erection problems</td>
<td>health.allrefer.com</td>
</tr>
<tr>
<td>women's urine blood</td>
<td><a href="http://www.familypractice.com">www.familypractice.com</a></td>
</tr>
<tr>
<td>book shop</td>
<td><a href="http://www.mondello.com">www.mondello.com</a></td>
</tr>
<tr>
<td>smolack</td>
<td><a href="http://www.innovate.com">www.innovate.com</a></td>
</tr>
<tr>
<td>web directory</td>
<td><a href="http://www.webdirectory.com">www.webdirectory.com</a></td>
</tr>
<tr>
<td>jobs</td>
<td><a href="http://www.cheaploans.com">www.cheaploans.com</a></td>
</tr>
<tr>
<td>military</td>
<td><a href="http://www.military.com">www.military.com</a></td>
</tr>
</tbody>
</table>

usage patterns
All of this is private.
Anything your users don’t want shared
We all store private information.
How well do we protect it?
Firewall!
Firewall!

VPN!
Firewall!

VPN!

Passwords!
Useless.
<table>
<thead>
<tr>
<th></th>
<th>OWASP Top 10 – 2010 (New)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Injection</td>
</tr>
<tr>
<td>A2</td>
<td>Cross-Site Scripting (XSS)</td>
</tr>
<tr>
<td>A3</td>
<td>Broken Authentication and Session Management</td>
</tr>
<tr>
<td>A4</td>
<td>Insecure Direct Object References</td>
</tr>
<tr>
<td>A5</td>
<td>Cross-Site Request Forgery (CSRF)</td>
</tr>
<tr>
<td>A6</td>
<td>Security Misconfiguration (NEW)</td>
</tr>
<tr>
<td>A7</td>
<td>Insecure Cryptographic Storage</td>
</tr>
<tr>
<td>A8</td>
<td>Failure to Restrict URL Access</td>
</tr>
<tr>
<td>A9</td>
<td>Insufficient Transport Layer Protection</td>
</tr>
<tr>
<td>A10</td>
<td>Unvalidated Redirects and Forwards (NEW)</td>
</tr>
</tbody>
</table>

Really Useless.

Encryption!
Hard.
Really hard.
We still need to protect it.
We have ideas.
Grendel!
Grendel!
A Secure Document Storage System
CAUTION

Simplest Thing That Can Work
CAUTION

Simplest Thing That Can Work
Minimum Viable Product
CAUTION

Simplest Thing That Can Work
Minimum Viable Product
Open Source
CAUTION

Simplest Thing That Can Work
Minimum Viable Product
Open Source
Big Plans for the Future
Simplest Thing That Can Work
Minimum Viable Product
Open Source
Big Plans for the Future
Gets the Fundamentals Right
Data Storage
Data Storage
Authentication
Data Storage
Authentication
Access Control
OpenPGP
OpenPGP

REST
OpenPGP
REST
Java + RDBMS
OpenPGP
Message Format

What PGP and GPG Use
OpenPGP Message Format

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This document is maintained in order to publish all necessary information needed to develop interoperable applications based on the OpenPGP Message Format.
Open Standard
Well Reviewed
Used Everywhere
Flexible
Confidentiality
Integrity
Keys!
Asymmetric
One Set Per User
Stored Encrypted with User’s Passphrase
Documents!
Arbitrary Contents
Recipients!
REST
HTTP
HTTP
Spoken Natively
Why REST?
Simple
Ubiquitous
Well-Understood
Easy to Debug
Lots of Free Features
OpenPGP

REST

Java + RDBMS
Java + RDBMS
Java 6
Java 6
Bouncy Castle
Java 6
Bouncy Castle
Jetty
Java 6
Bouncy Castle
Jetty
Jersey
Java 6
Bouncy Castle
Jetty
Jersey
Guice
Java 6
Bouncy Castle
Jetty
Jersey
Guice
Hibernate
Why Java?
Fast
Fast
Stable
Fast
Stable
Well-Understood
Why RDBMS?
You Already Have One
OpenPGP ✓
REST ✓
Java + RDBMS ✓
Easy To Use
One Config File
hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
hibernate.connection.username=grendel
hibernate.connection.password=sn00persn33krit
hibernate.connection.url=jdbc:mysql://db1.example.com/grendel_prod
hibernate.c3p0.min_size=10
hibernate.c3p0.max_size=50
$ java -jar grendel.jar schema \
   -c database.properties
create table documents (  
    name varchar(255) not null,
    body longblob not null,
    content_type varchar(40) not null,
    created_at datetime not null,
    modified_at datetime not null,
    version bigint not null,
    owner_id varchar(255) not null,
    primary key (name, owner_id)
) ENGINE=InnoDB;

create table links (  
    user_id varchar(255) not null,
    document_name varchar(255) not null,
    document_owner_id varchar(255) not null,
    primary key (user_id, document_name, document_owner_id)
) ENGINE=InnoDB;

create table users (  
    id varchar(255) not null,
    created_at datetime not null,
    keyset longblob not null,
    modified_at datetime not null,
    version bigint not null,
    primary key (id)
) ENGINE=InnoDB;
Running the Server
$ java -jar grendel.jar server \
   -c database.properties \
   -p 8080
Grendel’s API

/users
/users/{id}
/users/{id}/documents
/users/{id}/documents/{name}
/users/{id}/documents/{name}/links
/users/{id}/documents/{name}/links/{otherid}
/users/{id}/linked-documents
/users/{id}/linked-documents/{otherid}/{name}
Creating a User
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{
  "id": "codahale",
  "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{
  "id": "codahale",
  "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{
    "id": "codahale",
    "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{
    "id": "codahale",
    "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{
    "id": "codahale",
    "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{   "id": "codahale",
    "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{
  "id": "codahale",
  "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{
  "id": "codahale",
  "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{
    "id": "codahale",
    "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{
   "id": "codahale",
   "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale

Generates a New Key Pair
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{
    "id": "codahale",
    "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale

Generates a New Key Pair
Encrypts The Private Key
Creating a User

POST /users/ HTTP/1.1
Content-Type: application/json

{  
   "id": "codahale",
   "password": "woowoo"
}

HTTP/1.1 201 Created
Location: http://example.com/users/codahale

Generates a New Key Pair
Encrypts The Private Key
Stores It with the User Record
Storing a Document
Storing a Document

PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29v

yay for me

HTTP/1.1 204 No Content
PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29v
yay for me

HTTP/1.1 204 No Content
Storing a Document

PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29v

yay for me

HTTP/1.1 204 No Content
Storing a Document

PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29v

yay for me

HTTP/1.1 204 No Content
Storing a Document

```
PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29vd29v

yay for me
```

HTTP/1.1 204 No Content
Storing a Document

PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29v

yay for me

HTTP/1.1 204 No Content
Storing a Document

PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29v
yay for me

HTTP/1.1 204 No Content
Storing a Document

PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29v

yay for me

HTTP/1.1 204 No Content
Storing a Document

PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29v

yay for me

HTTP/1.1 204 No Content

Decrypts the User’s Key Set
Storing a Document

PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29vd29v

yay for me

HTTP/1.1 204 No Content

Decrypts the User’s Key Set
Signs the Document With It
Storing a Document

PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29v

yay for me

HTTP/1.1 204 No Content

Decrypts the User’s Key Set
Signs the Document With It
Encrypts the Document With It
Storing a Document

PUT /users/codahale/documents/foo.txt HTTP/1.1
Content-Type: text/plain
Authorization: Basic Y29kYWhhbGU6d29vd29v

yay for me

HTTP/1.1 204 No Content

Decrypts the User’s Key Set
Signs the Document With It
Encrypts the Document With It
Stores The Encrypted Document
Listing Documents
Listing Documents

GET /users/codahale/documents/ HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
  "documents": [
    {
      "name": "foo.txt",
      "uri": "/users/codahale/documents/foo.txt"
    }
  ]
}
GET /users/codahale/documents/ HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
  "documents":[{
    "name":"foo.txt",
    "uri":"/users/codahale/documents/foo.txt"
  }]
}
GET /users/codahale/documents/ HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{  
    "documents": [{
        "name": "foo.txt",
        "uri": "/users/codahale/documents/foo.txt"
    }]
}
Listing Documents

GET /users/codahale/documents/ HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
  "documents": [
    {
      "name": "foo.txt",
      "uri": "/users/codahale/documents/documents/foo.txt"
    }
  ]
}
Listing Documents

GET /users/codahale/documents/ HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
   "documents":[{
      "name":"foo.txt",
      "uri":"/users/codahale/documents/foo.txt"
   }]
}

Listing Documents

GET /users/codahale/documents/ HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
  "documents": [{
    "name": "foo.txt",
    "uri": "/users/codahale/documents/foo.txt"
  }]
}
Listing Documents

GET /users/codahale/documents/ HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
   "documents": [
      {
         "name": "foo.txt",
         "uri": "/users/codahale/documents/bar.txt"
      }]
}

Listing Documents

GET /users/codahale/documents/ HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
  "documents": [
    {
      "name": "foo.txt",
      "uri": "/users/codahale/documents/documents/foo.txt"
    }
  ]
}

Decryts the User's Key Set
Listing Documents

GET /users/codahale/documents/ HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
    "documents": [
        {
            "name": "foo.txt",
            "uri": "/users/codahale/documents/documents/foo.txt"
        }
    ]
}

Decrypts the User’s Key Set
Loads the List of Documents
Viewing a Document
Viewing a Document

```
GET /users/codahale/documents/foo.txt HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 200 OK
Cache-Control: private, no-cache, no-store
Content-Type: text/plain

yay for me
```
Viewing a Document

GET /users/codahale/documents/foo.txt HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 200 OK
Cache-Control: private, no-cache, no-store
Content-Type: text/plain

yay for me
Viewing a Document

GET /users/codahale/documents/foo.txt HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 200 OK
Cache-Control: private, no-cache, no-store
Content-Type: text/plain

yay for me
Viewing a Document

GET /users/codahale/documents/foo.txt HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 200 OK
Cache-Control: private, no-cache, no-store
Content-Type: text/plain

yay for me
Viewing a Document

GET /users/codahale/documents/foo.txt HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 200 OK
Cache-Control: private, no-cache, no-store
Content-Type: text/plain

yay for me
Viewing a Document

GET /users/codahale/documents/foo.txt HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 200 OK
Cache-Control: private, no-cache, no-store
Content-Type: text/plain

yay for me
Viewing a Document

GET /users/codahale/documents/foo.txt HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 200 OK
Cache-Control: private, no-cache, no-store
Content-Type: text/plain

yay for me

Decrypts the User’s Key Set
Viewing a Document

GET /users/codahale/documents/foo.txt HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 200 OK
Cache-Control: private, no-cache, no-store
Content-Type: text/plain

yay for me

Decrypts the User’s Key Set
Decrypts the Document
Viewing a Document

GET /users/codahale/documents/foo.txt HTTP/1.1
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 200 OK
Cache-Control: private, no-cache, no-store
Content-Type: text/plain

yay for me

Decrypts the User’s Key Set
Decrypts the Document
Verifies the Signature
Linking a Document to Another User
Linking a Document

PUT /users/codahale/documents/foo.txt/links/samquigley
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 204 No Content
PUT /users/codahale/documents/foo.txt/links/samquigley
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 204 No Content
Linking a Document

PUT /users/codahale/documents/foo.txt/links/samquigley
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 204 No Content
Linking a Document

PUT /users/codahale/documents/foo.txt/links/samquigley
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 204 No Content
Linking a Document

PUT /users/codahale/documents/footxt/links/samquigley
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 204 No Content
Linking a Document

PUT /users/codahale/documents/foo.txt/links/samquigley
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 204 No Content

Decrypts the Document
Linking a Document

PUT /users/codahale/documents/foo.txt/links/samquigley
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 204 No Content

Decrypts the Document
Re-encrypts the Document for Both Users
PUT /users/codahale/documents/foo.txt/links/samquigley
Authorization: Basic Y29kYWhhbGU6d29vd29v

HTTP/1.1 204 No Content

Decrypts the Document
Re-encrypts the Document for Both Users
Gives Read-Only Access To The Second User
GPG as a Service
GPG as a Service

(GaaS)
What’s the big deal?
Self-Defending Data
Self-Defending Data

sudo for the Web
Self-Defending Data

sudo for the Web

Privacy Wall
Self-Defending Data
Data Enforces Access Rules
Enforce Business Logic with Math
But Wait!
There’s More!
Authentication Done
Right
Adaptive Hashing
Adaptive Hashing
Centralized, as a Service
Adaptive Hashing
Centralized, as a Service
Resistant to Modern Attacks
sudo for the Web
Long-Lived Session Cookies
Long-Lived Session Cookies
Re-Auth for Privileged Access
Long-Lived Session Cookies
Re-Auth for Privileged Access
Mitigates XSS/CSRF (sorta)
Privacy Wall
You’re Locked Out
You’re Locked Out
This is Good
You're Locked Out
This is Good
Protects Everyone from Insiders
You're Locked Out
This is Good
Protects Everyone from Insiders
Protects Everyone from Outsiders
Summary
Grendel: OpenPGP + REST + Java
Grendel: OpenPGP + REST + Java
Self-Defending Data
Authentication + sudo
Grendel: OpenPGP + REST + Java
Self-Defending Data
Authentication + sudo
Privacy Wall
<table>
<thead>
<tr>
<th>OWASP Top 10 – 2010 (New)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 – Injection</td>
</tr>
<tr>
<td>A2 – Cross-Site Scripting (XSS)</td>
</tr>
<tr>
<td>A3 – Broken Authentication and Session Management</td>
</tr>
<tr>
<td>A4 – Insecure Direct Object References</td>
</tr>
<tr>
<td>A5 – Cross-Site Request Forgery (CSRF)</td>
</tr>
<tr>
<td>A6 – Security Misconfiguration (NEW)</td>
</tr>
<tr>
<td>A7 – Insecure Cryptographic Storage</td>
</tr>
<tr>
<td>A8 – Failure to Restrict URL Access</td>
</tr>
<tr>
<td>A9 – Insufficient Transport Layer Protection</td>
</tr>
<tr>
<td>A10 – Unvalidated Redirects and Forwards (NEW)</td>
</tr>
</tbody>
</table>

Progress

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
Future Directions
Sessions
Sessions

OAuth (2.0)
Sessions

OAuth (2.0)

Spreading the Ideas
Questions?

http://github.com/wesabe/grendel