# Automated Certificate Management

...

## ACME + Let's Encrypt

Richard Barnes <rbarnes@mozilla.com>
@rlbarnes

# What problem are we trying to solve?

Not enough use of encryption in applications

- **~40%** of Firefox pageloads are HTTPS
- **~64%** of Firefox HTTP transactions are HTTPS
- **~62%** of emails received by Gmail are over STARTTLS
- Routers? Home gateways?

These numbers should be **100%**

# Getting a certificate is no fun

*"I can't f'ing figure out how to get a cert from [redacted] - kid you not...*

*god help people that don't know what a CSR is...*

*I am like 45 minutes in"*

— Cullen Jennings, PhD
Cisco Fellow
Former IETF Area Director

# Let’s do DHCP for certificates

# Initial efforts

SSLMate - Consistent REST API to a bunch of existing CAs

CertSimple - Semi-automated EV certificates

Let's Encrypt - A new CA only accessible through a REST API

*(more on Let's Encrypt in a moment)*

# ACME

"Automated Certificate Management … Environment?"

Goal: One REST API that all CAs can use

> … so that it makes sense to build tooling into things like web servers

Currently implemented by Let's Encrypt

There's an IETF working group

And an Internet-Draft

And you can contribute on Github

# Getting a certificate with ACME

1. Make an account
2. Prove that you own some domains
3. Issue a certificate for that domain

# Make an account

Create a key pair – this will be the “password” for your account

(All future messages to the server will be signed with this key pair)

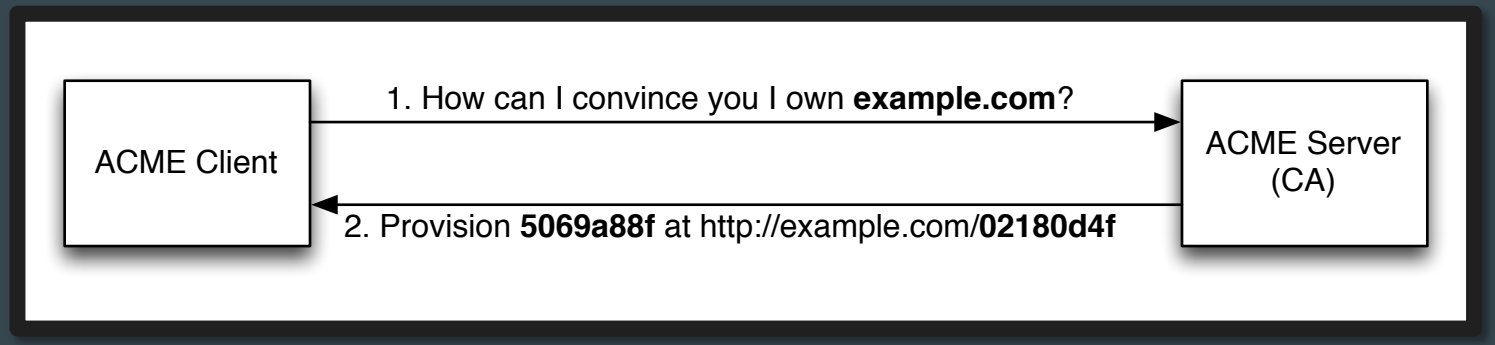
Register the key pair with the CA, along with contact info

# Prove you own some domains (1)

First you tell the server that you would like to be authorized for **example.com**

The server will ask you to prove that you control **example.com**

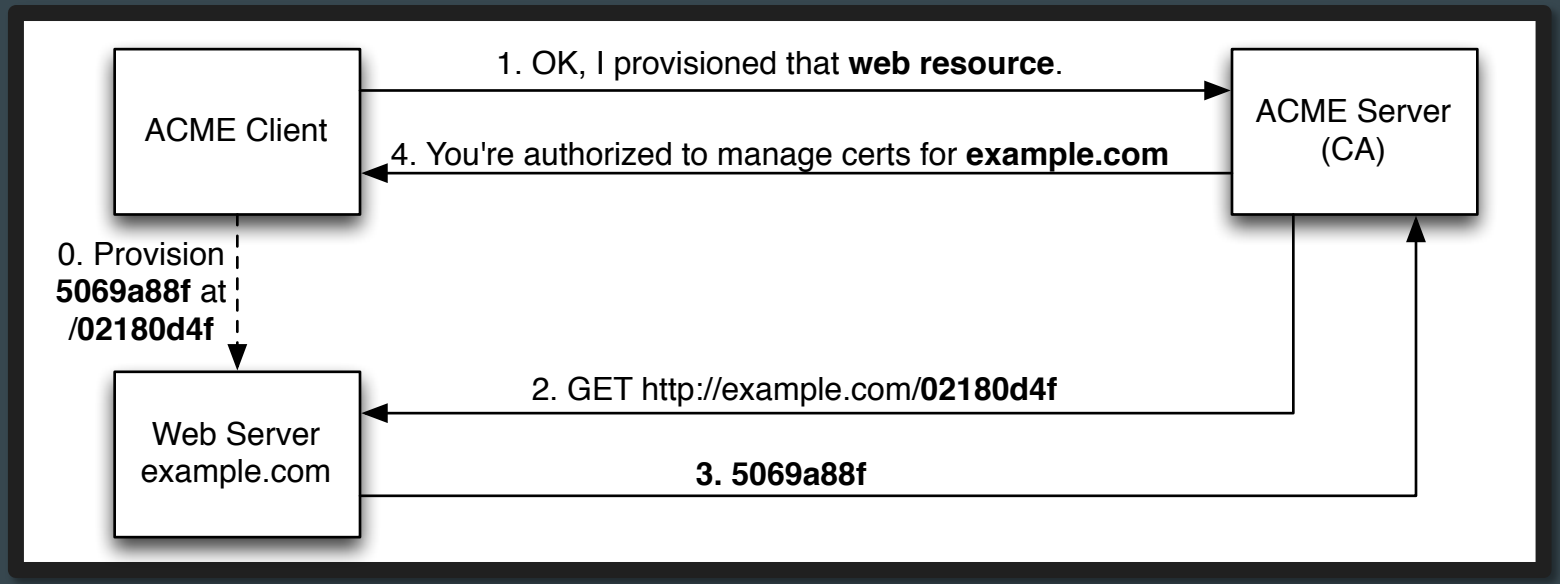
… by completing a **challenge**, like provisioning a file on **http://example.com**



# Prove you own some domains (2)

Once you fulfill the challenge, you let the server know, and the server will check

If the expected file is there, your account is now **authorized for example.com**

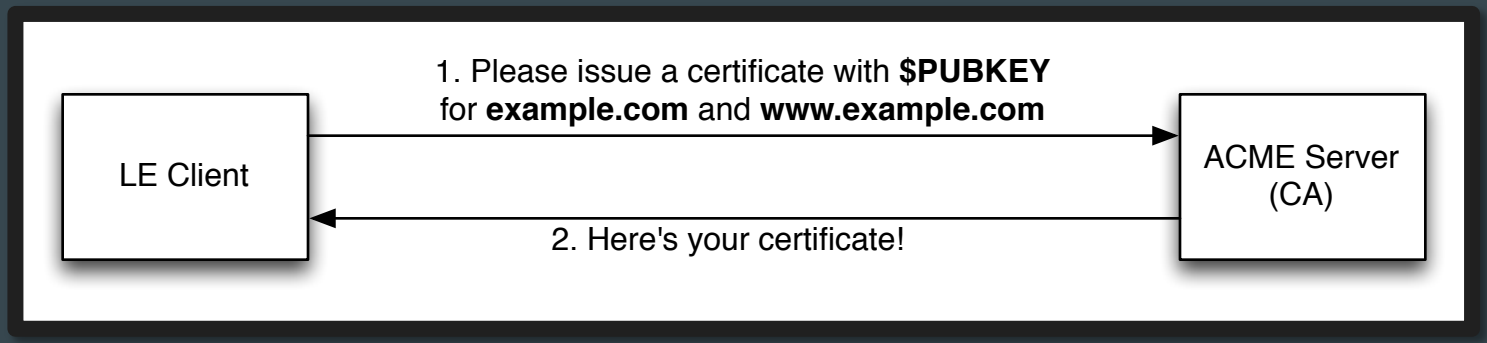


# Issue a certificate

Say you're authorized for some names, like **example.com** and **www.example.com**

Then you can make a certificate just by making a **Certificate Signing Request (CSR)**

… and sending it to the CA



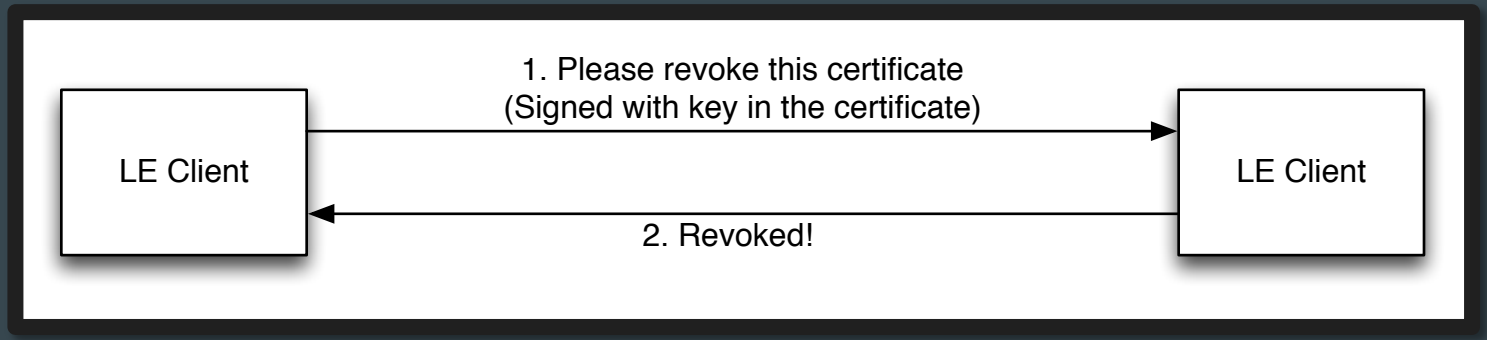
# Register → Authorize → Issue

Three steps, all done with HTTP requests to the CA

There's a little bit of signing magic to authenticate the client to the CA …

… but other than that, it's just JSON over HTTP

# And if anything should go wrong…




Let's
Encrypt

# Let’s Encrypt is a new CA

- Free - Non-profit, funded by sponsors
- Automatic - The only way to get a certificate is through the API (ACME)
- Secure - All SHA-2 from the start, ECDSA coming soon (also, no humans in the loop)
- Transparent - All certificates logged to CT; public metrics
- Open - All code is open source
- Let’s Encrypt root CA is already in all your browsers!

# How to get a certificate

General availability on December 3

Until then, sign up for the beta

# How to get a certificate

General availability on December 3

Until then, sign up for the beta

Use the futuristic* official client!


** Where "futuristic" == "works OK about 75% of the time; requires root; probably breaks your nginx; ..."*

# How to get a certificate

General availability on December 3

Until then, sign up for the beta

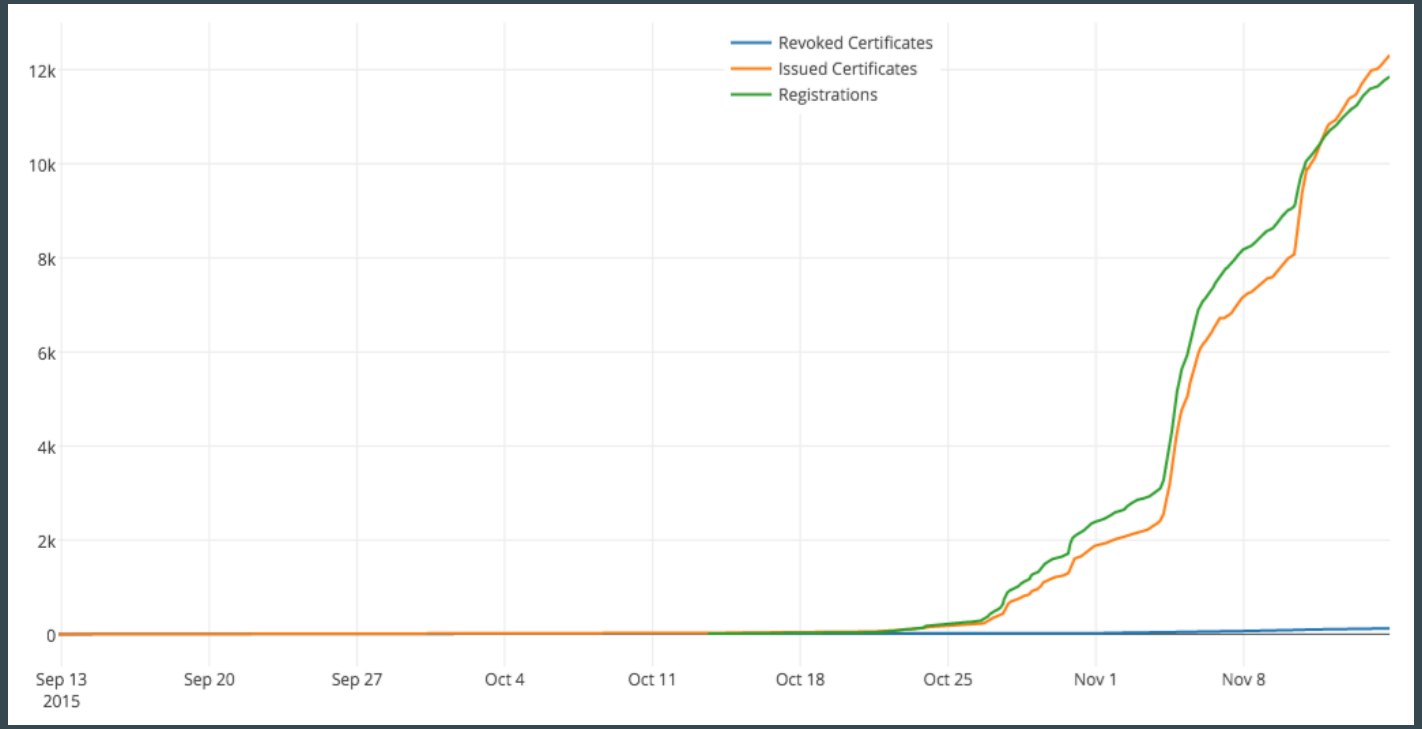
Use the futuristic* client software!

… or one of the various other clients that other people have written

… or use a server / hosting provider that has it integrated (e.g., Akamai)

… or write your own!

# Up and to the right!



# Security by Default

Everything we do on the net needs security

Security needs to be automatic

Let's Encrypt on web servers is a good start

We need to apply this automation to more CAs and more things

What is still running unsecured in your network?