

# Apache Milagro (incubating)

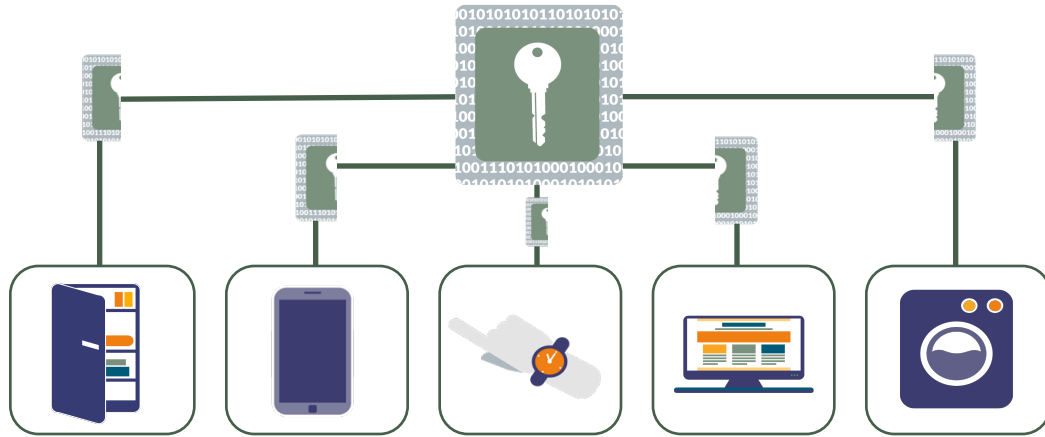
An Introduction  
ApacheCon North America



Apache Milagro will establish a  
new independent security  
framework for the Internet

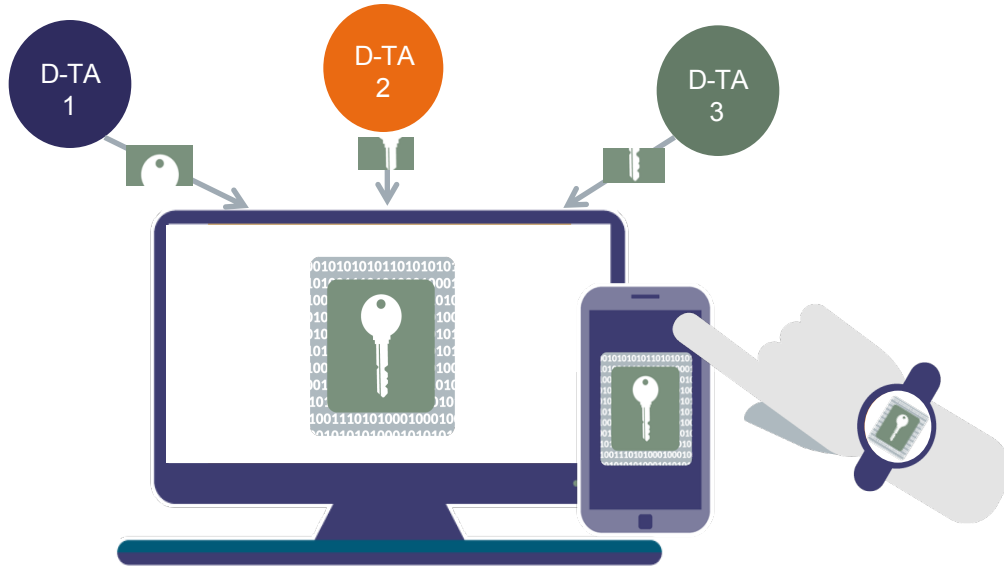


# A Distributed Cryptosystem



**Secure the Future of the Internet**

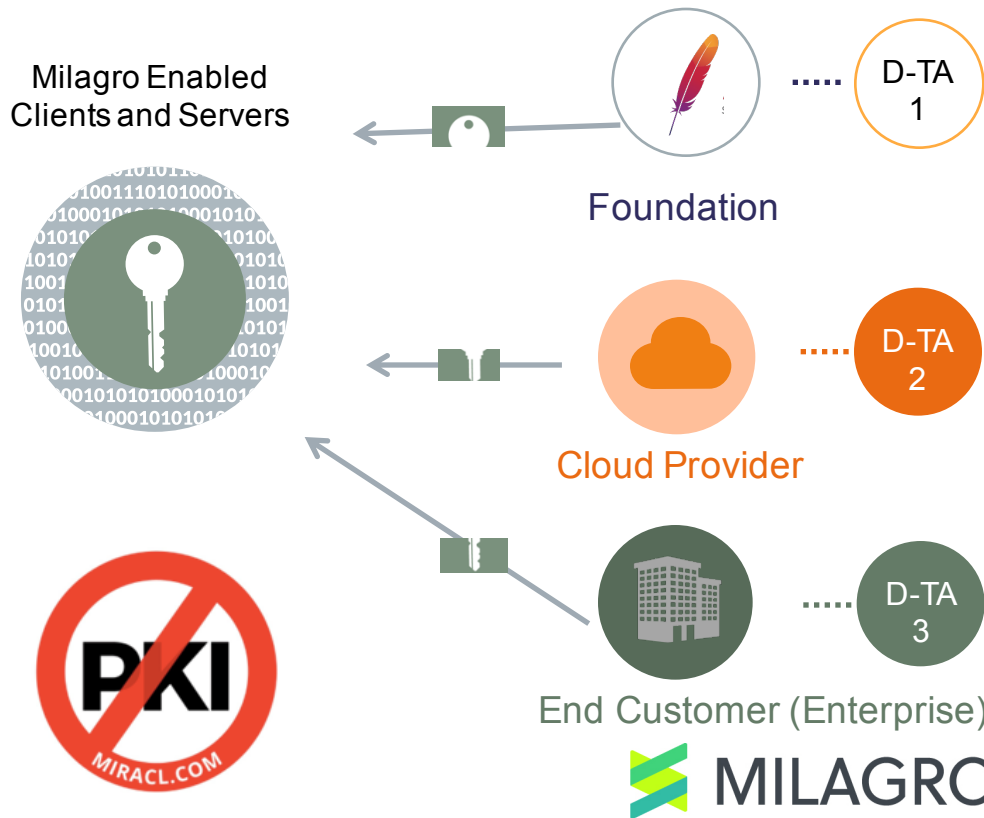
# Distributed Trust Ecosystem



-Milagro enabled apps and things receive their **key shares, or fractions**, from Distributed Trust Authorities.

-Keys have Identity "burned in"

# Distributed Trust Authorities



- Anyone or organization can become a Distributed Trust Authority

- And run it in any geography or jurisdiction

- There is no PKI 'root' – the future is decentralized

# Milagro Multi-Factor Authentication

Eliminates  
the risk of password  
database breach

| Username              | Password    |
|-----------------------|-------------|
| peter.black@gmail.com | Password123 |
| s.harrow@yahoo.co.uk  | fluffy      |
| ted.yipp@business.com | paris       |
| smith@company.com     | secure1     |



Improves  
authentication / signature  
user experience



Improves  
authentication security  
to multi-factor



Identity based cryptographic multi-factor authentication and digital signature protocol that replaces passwords.

Milagro MFA runs entirely in software – it's browser / app friendly.



# Milagro TLS Library

Non-interactive:  Interactive:  
**authenticate clients via a digital signature**      **create certificate-less TLS with forward secrecy**



The same protocol run interactively creates an authenticated key agreement between client & server or peer to peer

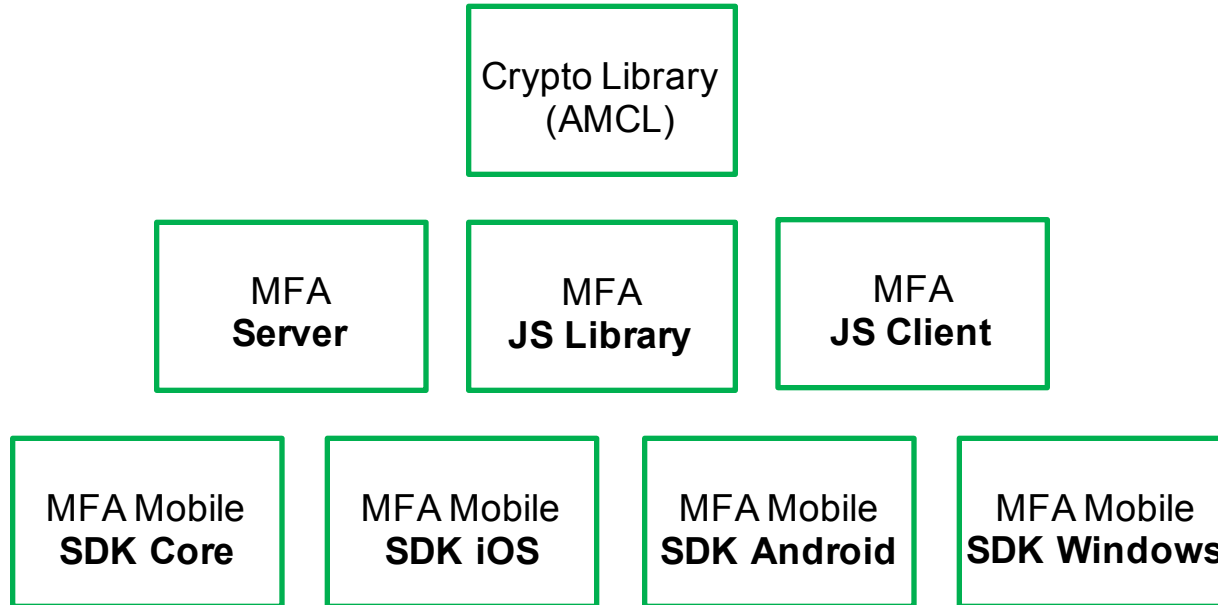


# About Milagro Multifactor Authentication





# A Toolkit for Multifactor Authentication



# Zero-Knowledge Proof Authentication Without Passwords

## Authentication into Web Applications

User registers in browser:  
receives a client secret and  
extracts a “PIN”

User logs in by using their PIN  
to recreate the client secret

1

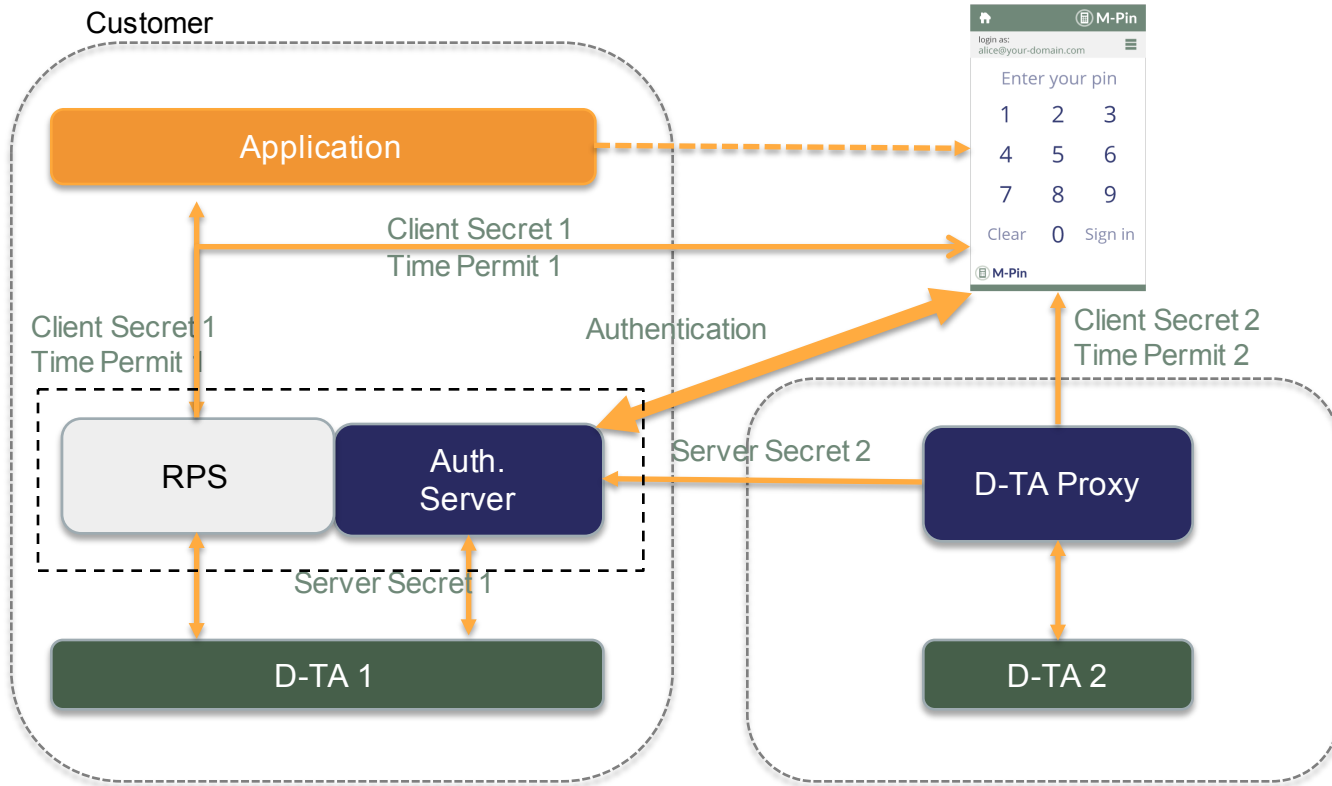
2

## Mobile Out-of-Band Authentication into Web Applications

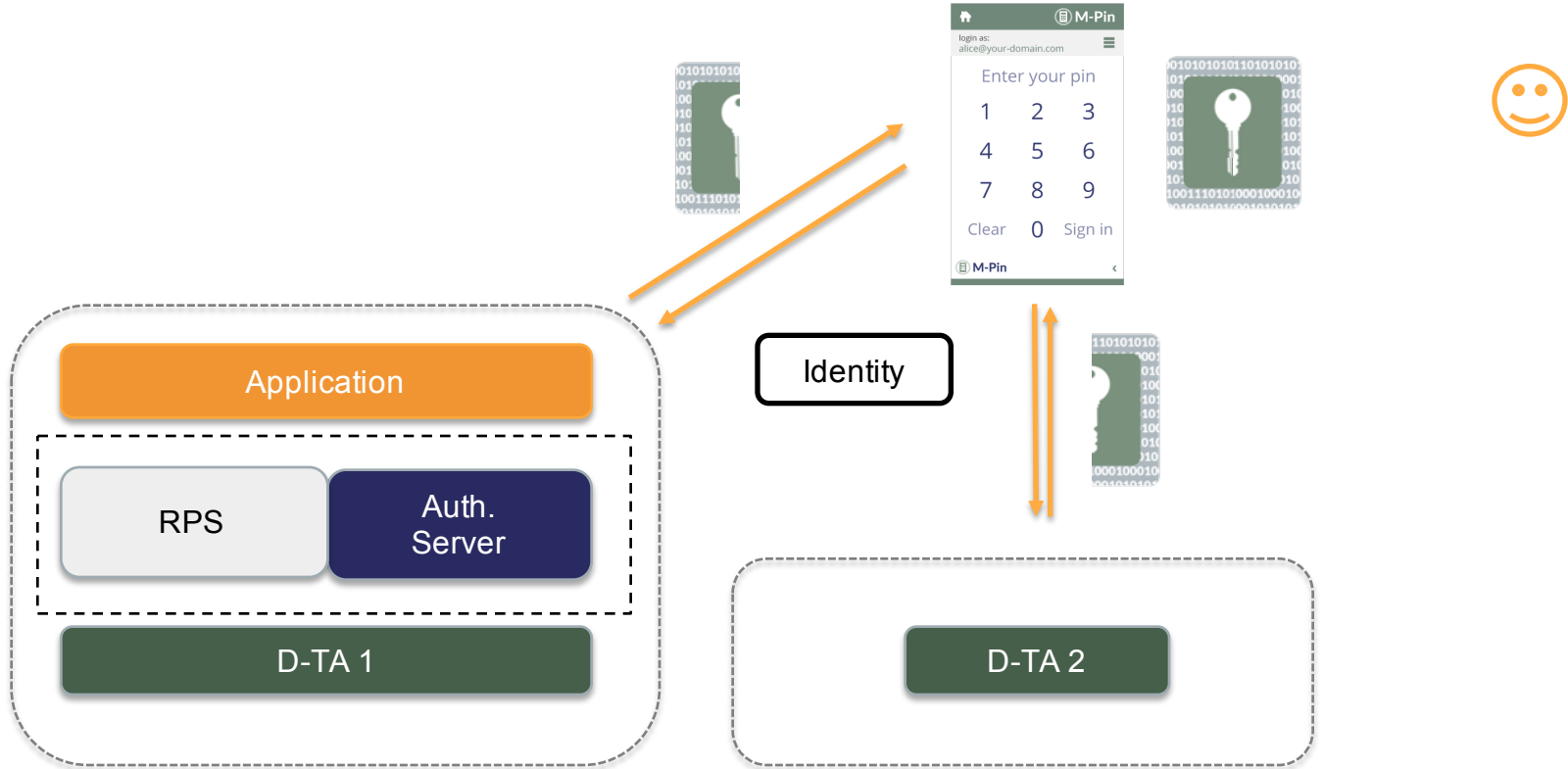
User registers on mobile:  
receives a client secret on  
mobile device, extracts a “PIN”

User logs in by entering an  
access code from the website  
and PIN into the mobile app

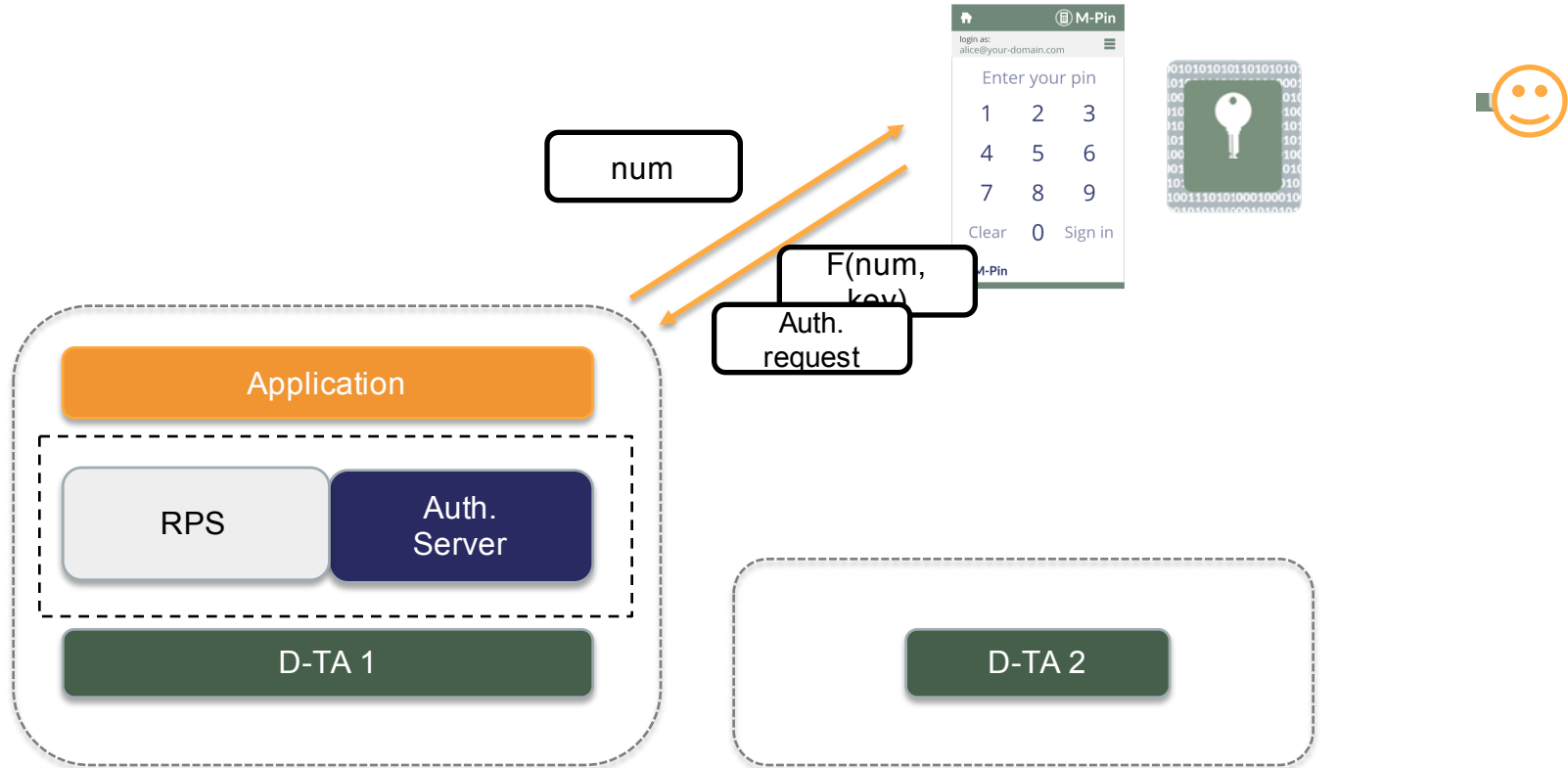
# Milagro MFA: How It Works



# Milagro MFA: Registration



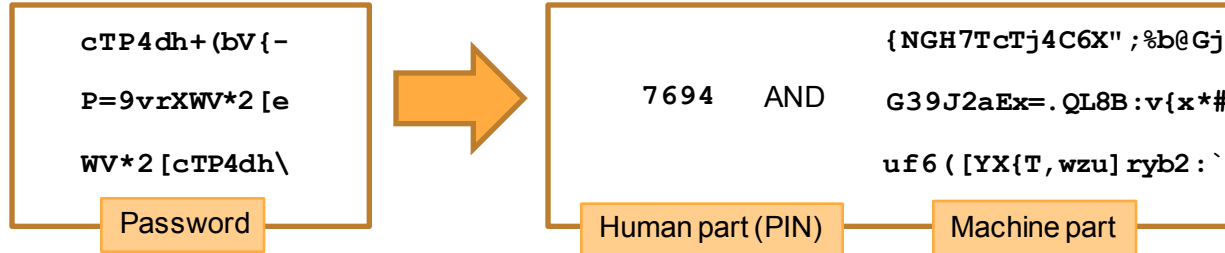
# Milagro MFA: Authentication



# Milagro MFA Integration



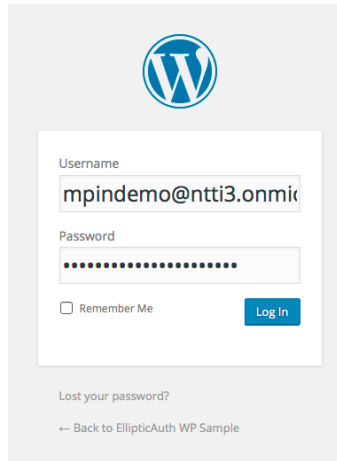
# Let The Machines Work For Human Comfort



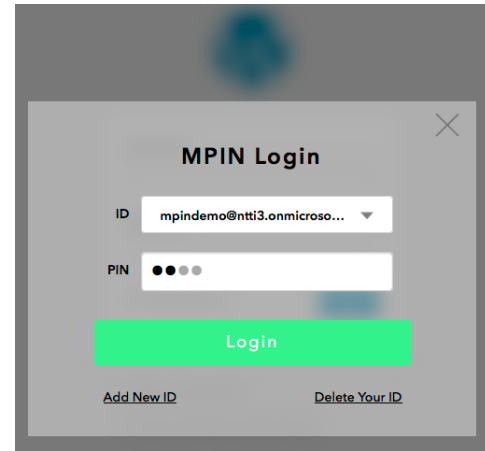
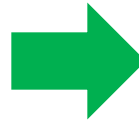
- At least 12 characters from upper-case and lower-case letters, and ...
- You must change it every 2 month.
- You must choose independently random passwords for all accounts.
- **4 digit number is OK for PIN.** Resiliency against brute force attacks.
- **You do not need to change secrets.** Zero-knowledge proof without credential database, hence no breach.
- **You may use the same PIN for all accounts.** Machine generates random OTP from the two factors, with your identity burned in.

# Demo: MFA on WordPress

Override the standard password login with Milagro-MFA without modifying the code.



The image shows the standard WordPress login form. At the top is the WordPress logo. Below it is a form with a 'Username' field containing 'mpindemo@ntti3.onmic', a 'Password' field with masked characters, a 'Remember Me' checkbox, and a blue 'Log In' button. At the bottom, there is a link for 'Lost your password?' and a link to '← Back to EllipticAuth WP Sample'.

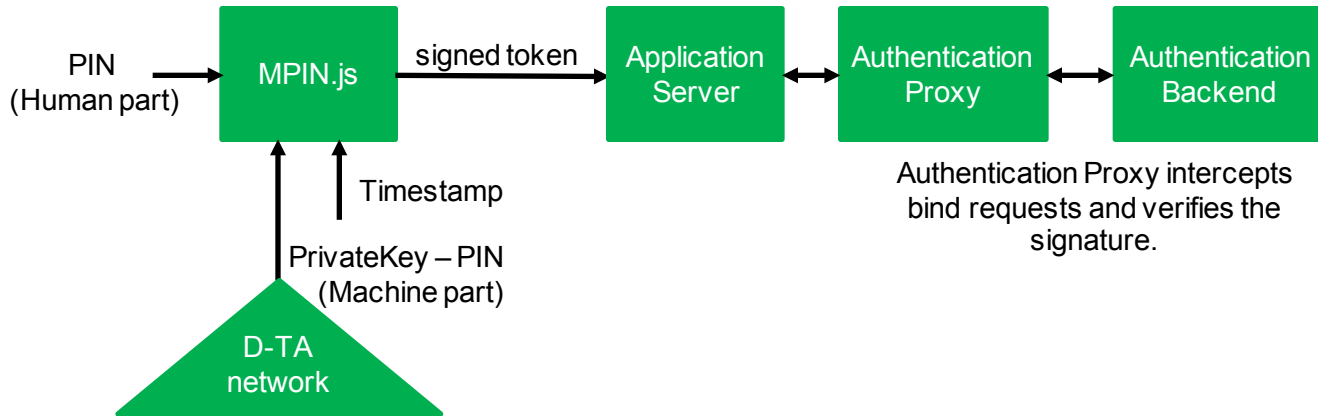


The image shows the Milagro-MFA 'MPIN Login' form. It has a title 'MPIN Login' and a close button (X) in the top right. Below the title is an 'ID' dropdown menu with the value 'mpindemo@ntti3.onmicro...'. Below that is a 'PIN' field with three masked characters. A large green 'Login' button is centered below the PIN field. At the bottom, there are two links: 'Add New ID' and 'Delete Your ID'.



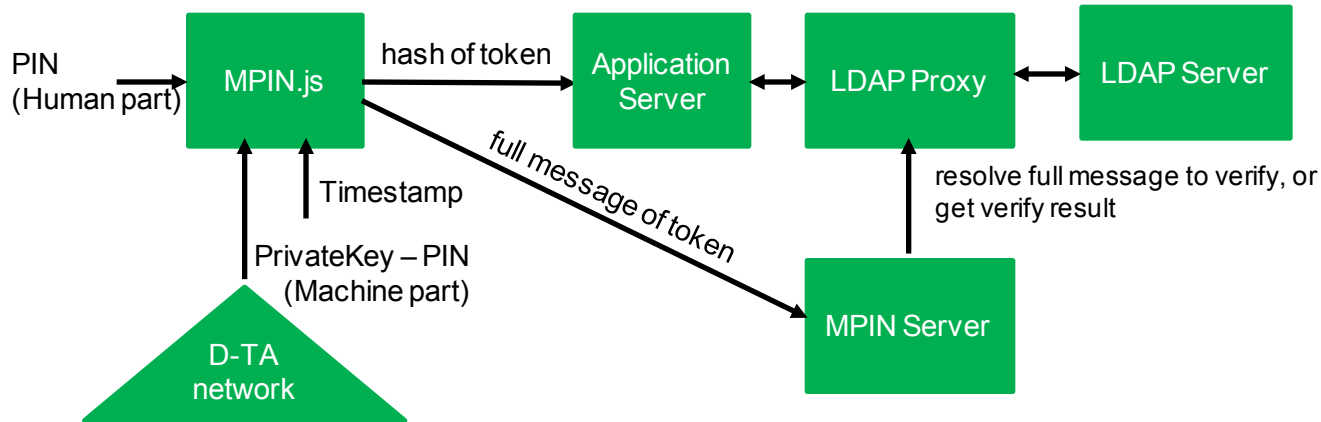
# Milagro: Plug-In Architecture

MPIN.js produces digitally signed one time token using PrivateKey, and submit it to the password form.



# Milagro: Implementation

MPIN.js communicates with MPIN server to submit full message of signed token.  
MPIN.js submits tokenized message (typically hash value) in the password form.



# Milagro: Easy Installation

MPIN.js overrides the standard password login form.

```
<!DOCTYPE html>
  <!--[if IE 8]>
    <html xmlns="http://www.w3.org/1999/xhtml" class="ie8" lang="en-US">
  <![endif]-->
  <!--[if !(IE 8)]><!--
    <html xmlns="http://www.w3.org/1999/xhtml" lang="en-US">
  <!--![endif]-->
  <head>
<!-- mpin -->
  <link href="https://public.milagro.io/public/css/mpin.min.css" rel="stylesheet">
  <script src="https://public.milagro.io/public/js/mpin.js"></script>
<!-- end mpin-->
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>EllipticAuth WP Sample &rsquo; Log In</title>
  <link rel="stylesheet" id="open-sans-css" href="https://fonts.googleapis.com/css?family=Open+Sans%3A300italic%3A%3A#038;ver=4.4.2" type="text/css" media="all" />
  <link rel="stylesheet" id="dashicons-css" href="https://iam.ellipticauth.com/apps/wordpress/wp-includes/css/dashicons.min.css" type="text/css" media="all" />
  <link rel="stylesheet" id="login-css" href="https://iam.ellipticauth.com/apps/wordpress/wp-admin/css/login.min.css" type="text/css" media="all" />
  <script>if("sessionStorage" in window){try{for(var key in sessionStorage){if(key.indexOf("wp-aut")>0){sessionStorage.removeItem(key)}}}catch(e){}}</script>
  <meta name="robots" content="noindex, follow" />
  </head>
  <body class="login login-action-login wp-core-ui locale-en-us">
<!-- mpin -->
  <div id="mpinClient" data-mpin-mode="displayed" data-mpin-pin-max-length="10" data-mpin-pin-min-length="4" data-mpin-image-base-url="https://public.milagro.io/public/images" class="mpin-hide" data-mpin-form-id="loginform">
<!-- end mpin-->
  <div id="login">
    <h1><a href="https://wordpress.org/" title="Powered by WordPress" tabindex="-1">EllipticAuth WP
    <p class="message"> You are now logged out.<br />
  </p>
```



# Milagro: Easy Installation

1. Import MPIN.js at the frontend.

```
<link href="https://public.milagro.io/public/css/mpin.min.css" rel="stylesheet">  
<script src="https://public.milagro.io/public/js/mpin.js"></script>
```

2. Insert LDAP proxy from Milagro between your target application and the LDAP server.



# About Milagro TLS Library



# Certificate Less TLS with Perfect Forward Secrecy

An extension to the ARM mbed TLS library (<https://tls.mbed.org/>)

Introduces two new key-exchange algorithms

Designed for Client-to-Server communication (MILAGRO\_CS)

Designed for Peer-to-Peer communications (MILAGRO\_P2P)

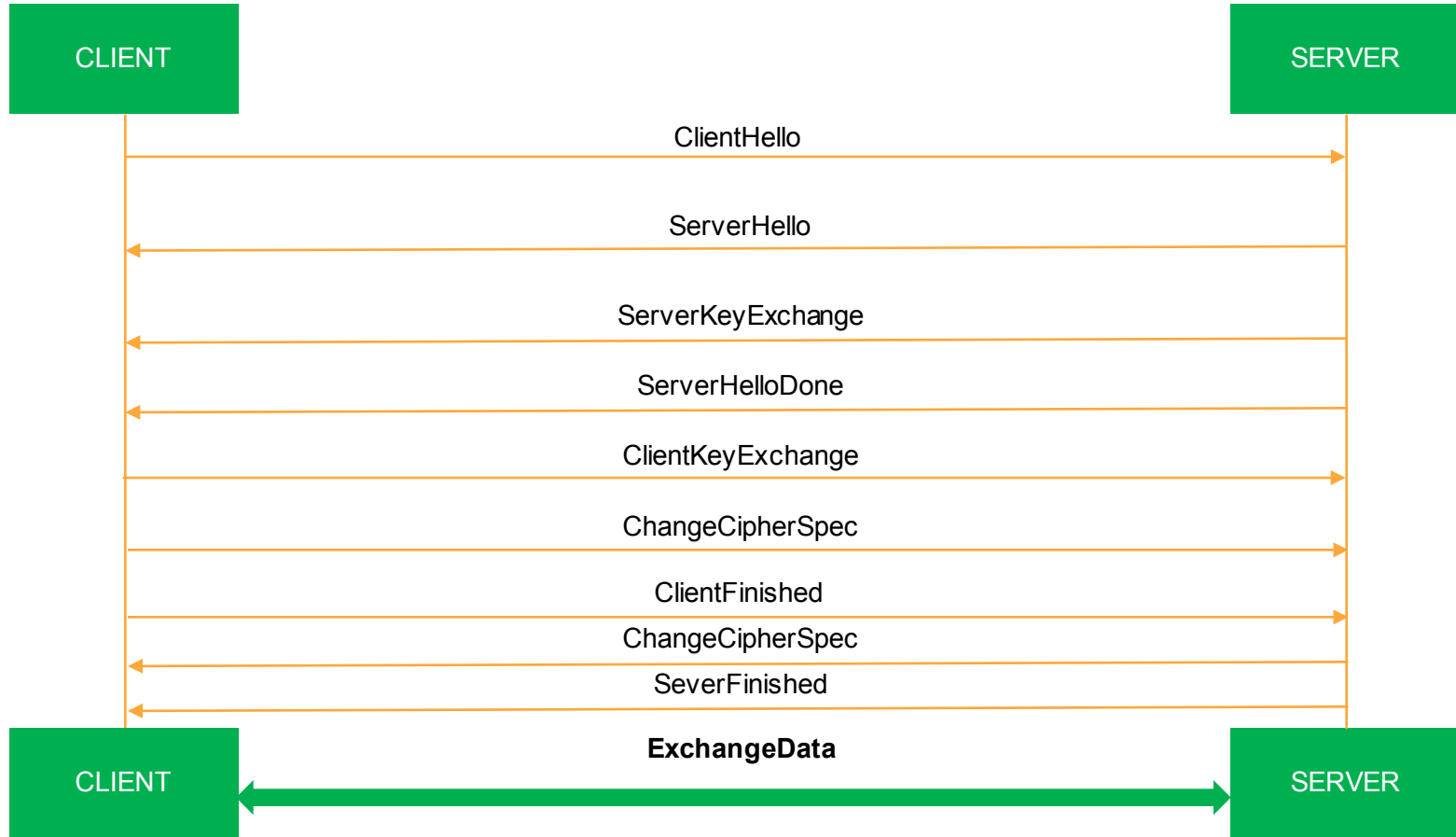


# New Environments Need New Solutions

| Current Convention                       | Milagro TLS                                     |
|--|---|
| Open SSL Complexity                      | Easily Auditable                                |
| Certificate Management Complexity in TLS | X-509 Digital Certificates Not Needed           |
| Certificate / Key Revocation             | Revocation That Works (time and identity based) |
| Limited Security for IoT & Containers    | Identity based C/S and P2P Channel Security     |

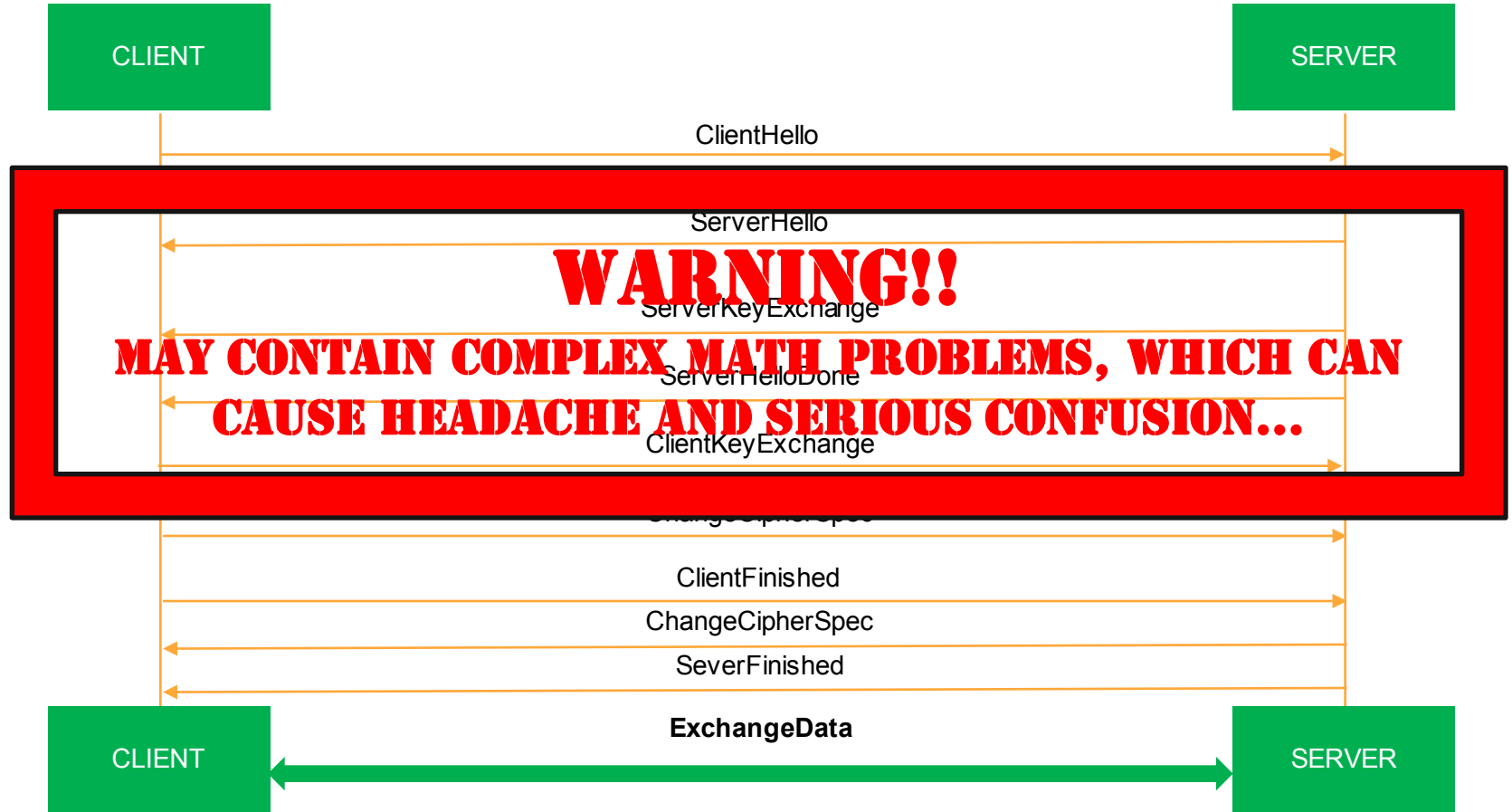


# Standard TLS





# Standard TLS



# MILAGRO / TLS

CLIENT

SERVER

## CLIENT AUTHENTICATION – step 1

We getting some values and place them in extension  
We are getting: time value, generate random number, we hashed  
Client identity etc.  
(even more complex MATH... 😊)

## GENERATING AES KEY



ClientHello + {A,U,V,t}

ServerHello

KeyExchange = {W}

ClientHelloDone

ServerKeyExchange = {R}

ClientCipherSpec

Finished

## SERVER AUTHENTICATION

## FORWARD SECRECY DIFFIE HELLMAN

## GENERATING AES KEY



$$VG1 = -(x+y)(s.CG1)$$

## FINAL CLIENT AUTHENTICATION

ExchangeData – identity

CLIENT

SERVER

# What Milagro TLS Can Deliver

Data Center  
Cryptosystem

Orchestration  
Host Security

Container 2  
Container

Websites and  
TLS w/o Certs

MQTT & CoAP

Distributed Trust

Pairing Based Cryptography (IBE)

Easy Revocation Through Invalidation

Certificate Authority or self signed certs not needed

Enables P2P TLS for devices and containers



# Pre-Alpha Code Available Now

Milagro – DTA code

Pre-Alpha: Extended mbedtls library

Pre-Alpha: P2P Wang / Chow-Choo library

Draft Milagro TLS White Paper

Demo at booth on CS & P2P

**WARNING: May Contain Nuts**



# What's Ahead



# Developing Milagro (draft)



**Q2'16**

Milagro MFA  
1.0

Milagro TLS  
0.1

**Q3'16**

Milagro TLS  
1.0

**Q4'16**

Distributed  
Cryptosystem  
for Datacenters  
0.1

**Q1'17**

Distributed  
Cryptosystem  
for Datacenters  
1.0





# MILAGRO

[milagro.incubator.apache.org](https://milagro.incubator.apache.org)



@ApacheMilagro