**Full title describing purpose:**

Avoiding insecure password transmission by using a local signature solution to digitally sign a random one-time challenge.

(A signature solution for authentication using Bitnation ID key pair)

**Problem statement:**

Sending passwords over internet connections is not without security risks. Most sites today use https to protect the user from man-in-the-middle attacks and hostile access nodes.

Those security measures are a good start, but most of the time the user still enters a password to authenticate with the server and/or site. Sending passwords should be avoided as far as possible, and that is the problem we address.

By using a local signature solution to digitally sign a random one-time challenge from the server, the user can prove to the server/site that they are in possession of the “password” - that is - the secret key.

**How to use the Bitnation ID keys for secure authentication.**

**Define BID keys:**

Bitnation ID contains an asymmetric key pair that can be used with the Bitnation Public Notary. By pasting their keys and entering password, the users digitally sign the document.

**Solution:**

Here we demonstrate how to use the key pair to authenticate by sending a unique signature to the server. This signature can only be verified with the public key matching the secret key that was used to create the signature. By using the public key much as a username, a server, or other client, can confirm that the person requesting access has in possession the matching secret key. The concept described is a manual process, but this can also be automated to create a “social sign in button”. A more direct use can be in-app digital signature of arbitrary data, document hashes or short texts.

**Technical approach:**

The keys are Elliptic Curve generated with javascript functions using ed22519 curve.

The signature process is ECDSA.

<https://docs.google.com/presentation/d/16J-_z7dlsYJCMnmsXLgDaUzuTFXXtbPcetKKBjdfS-0/edit#slide=id.p>

**Step description:**

Step 1: The **user selects the ID key pair** to use for authenticating or signing.

Step 2: **Client quietly supplies the public key**. Server assumes **public key is the user**.

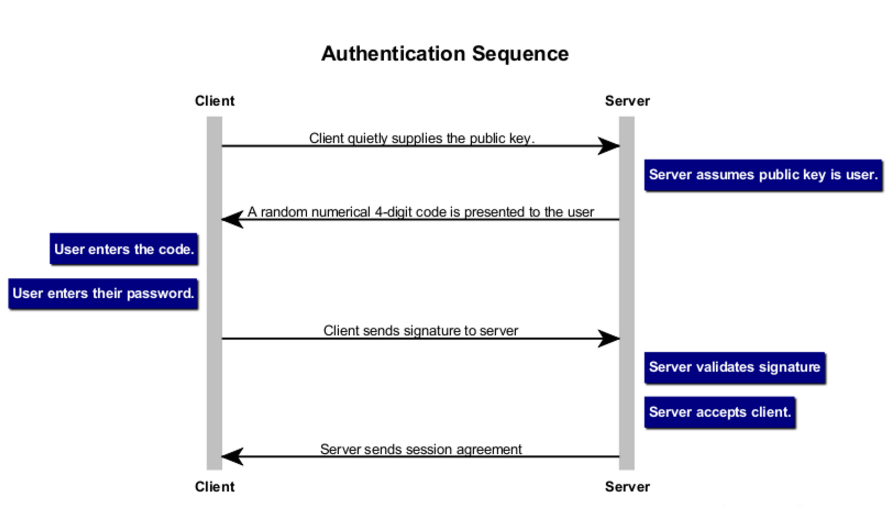
Step 3: A random numerical **4-digit code** is presented to the user, by the server.

Step 4: The **user** **enters the number** shown on screen. This is stored for next step.

Step 5: **User enters the password** to decrypt the private key.

Step 6: **Client sends signature to server**. Server uses the public key to validate.

Step 7: **Server accepts client**. The end.



A signature solution for authentication

using Bitnation ID key pair

Select the keys

We assume the user has created their Bitnation ID with keypair.

The private key is encrypted, so a password is always needed to unlock it for any signatures. This is the password we use in the last step.

Step 1: The user selects the ID keypair to use for authenticating or signing.

Client gives public key to server

The user now connects to a server that supports Bitnation ID authentication.

This could also be another client, P2P, but that is another story.

Step 2: User quietly supplies the public key. Server assumes key is the user.

Server displays a random PIN value

The server must now make sure the user is actually the holder of this keypair.

Step 3: A random numerical 4-digit code is presented to the user.

User enters PIN shown by server

Step 4: Well, the client enters the number shown on screen. This is stored for next step.

User enters password for private key

We now have the number from the server, and we need to sign it to return a signature to the server for validation of keypair.

Step 5: User enters the password to decrypt the private key.

Server validates using public key and PIN value.

After the user has entered their password, a signature is created with the private key. This is returned to the server, and validated using the public key.

Step 6: Client sends signature to server. Server uses the public key to validate.

ALL OK!

The server has now made sure the client has control over the private key, and accepts the client is the same as the public key provided.

Step 7: Server accepts client. The end.

Authentication Sequence for Web

Client->Server: Client quietly supplies the public key.

note right of Server: Server assumes public key is user.

Server->Client: A random numerical 4-digit code is presented to the user

note left of Client: User enters the code.

note left of Client: User enters their password.

Client->Server: Client sends signature to server

note right of Server: Server validates signature

note right of Server: Server accepts client.

Server->Client: Server sends session agreement

