Using YubiKey as a One-Time-Password Token

Purpose
This document provides directions to program a YubiKey as a one-time-password token for use with Duo Two-Factor Authentication (2FA) at the University of Washington.

Prerequisites
- Eligibility to use Duo two-factor authentication (2FA)
- YubiKey Personalization Tool (https://www.yubico.com/products/services-software/personalization-tools/use/) installed
- Possession of a compatible YubiKey (cannot program the Security Key, which only uses FIDO U2F and directions will be different with a YubiKey VIP key)

First Steps
- Decide which algorithm to use. Duo supports YubiKey AES and OATH-HOTP. A helpful comparison between YubiKey AES and OATH-HOTP is here.
- Note: The mainframe system (Keynes) cannot support longer than 8 character passcodes. If you plan on using your YubiKey to sign in to the mainframe (Keynes) you'll need to use the OATH-HOTP algorithm and configuration directions.
- If you're planning on using the same secrets for other external services that rely on Yubico's validation servers you'll need to use the Yubi Key AES Configuration algorithm and configuration directions.

Directions
- YubiKey OATH-HOTP Configuration
- YubiKey AES Configuration

⚠️ Configuring your YubiKey as a one-time-password token for Duo by following these directions will overwrite your current YubiKey configuration.

If you currently use your YubiKey with other services (e.g. LastPass), you can do one of two things:

1. Configure your new Duo secrets in 'Configuration Slot 2' (long press of the button) by using slot 2 anywhere where this documentation specifies slot 1. OR
2. Use the YubiKey AES Configuration documentation as written and ensure you follow the steps to upload your new public key, private key, and secret to Yubico. You'll need to re-configure your existing services to use your newly generated secrets so make sure you're already authenticated with them before you wipe away the configuration.

Note: UW-IT does not recommend option 2 since it has potential security ramifications. If a one-time-passcode were to be stolen somehow, that passcode has the potential to be used on another service that relies on the same secret. For that reason, we recommend (some variation on) option 1.