Error-tolerant authentication
Improving usability of text-based passwords via keystroke dynamics

Motivation
• Text-based passwords remain as the most prevalent authentication factor on online applications
• Passwords impose high cognitive burden on users (vast amount of services with strict & incoherent security policies)
• Increased security usually lowers usability → users circumvent the security requirements
• Could we increase usability without the loss of security?

Objective
• Use biometric user verification via keystroke dynamics during password login
• Apply error-correction on the password input based on biometric confidence
• Authenticate user without prompting to retry the login process

Hypothesis
• Less rejected login attempts to improve usability
• Lost password entropy feasibly supplemented by biometrics
• Non-intrusive from the user’s perspective
• Applicable to online settings

Current progress
• Running prototype ready
• Dataset collection underway (login attempts with keystroke timing data)

Concept

Normal password authentication
1. User inputs password with errors
2. Login submitted to server
3. Password hash mismatch → access rejected

Proposed solution
1. User inputs password with errors
   • Key event timings recorded during typing (biometric sample)
2. Login + keystroke timing data submitted to server
   • Biometric sample verified against user template
   • Apply error-correcting code on input to attempt retrieval of original password
3. Password hash matches → access granted

Steps
• Registration
  • Password is encoded with Reed-Solomon code with varying error-correction capacity \( t = \{1, 2, 3\} \)
  • Parity bytes stored separately from password hash
• Initialization
  • First \( n \) logins collect keystroke dynamics template
• Authentication
  • Keystroke dynamics timing data recorded
  • Biometric features extracted from the data
  • Classifier result to yield a confidence level
  • Corresponding error-correction applied to password