

A library in C++ for Graphical Password and Human Verification

Sagar Limaye, Pulkit Biyani, Vaibhavi Moghe

Abstract

A graphical authentication method is implemented through selection of discretized points on an image. Discretization is used in click-based graphical passwords so that approximately correct entries can be accepted by the system. For human verification, we make use of a grid of images where users are asked to select certain images matching a common feature or description, as an alternative against text-based verification. This is a library of a security system, which provides an API for authentication and human verification.

Introduction

- Secret that a user inputs to a computer with the aid of the computer's graphical input
- **Recall-based:** Users can draw out their password on a grid or canvas
- **Recognition-based:** Users memorize a set of images during password creation and recognize them from decoys during log in
- Alternative to text-based
- Human verification is the user selecting a set of images matching some description given (image CAPTCHA)

Requirements

- Cross platform app development framework
Qt used to build the library
- Works on x86 Windows and Ubuntu



Background and motivation

- Flexible password scheme – does not require predefined click regions
- Image must be rich and intricate enough so that many possible points are available
- Five or six click points has more possible password combinations than standard 8 character text password [1]

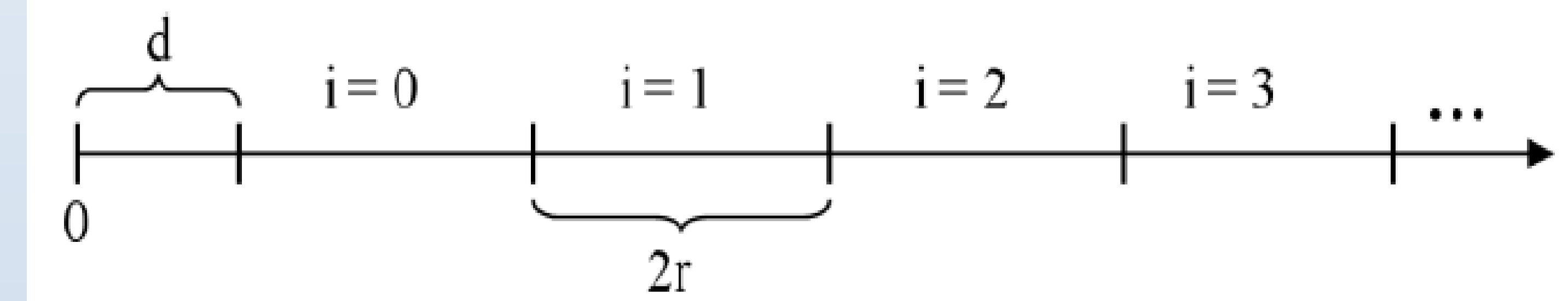
Method	Issue
Click based	Limited password space
Persuasive click points	Too much precision required
Static grid based	Edge problem
Robust discretization	False accepts and false rejects
Centered discretization	Minor usability issues

Problem statement

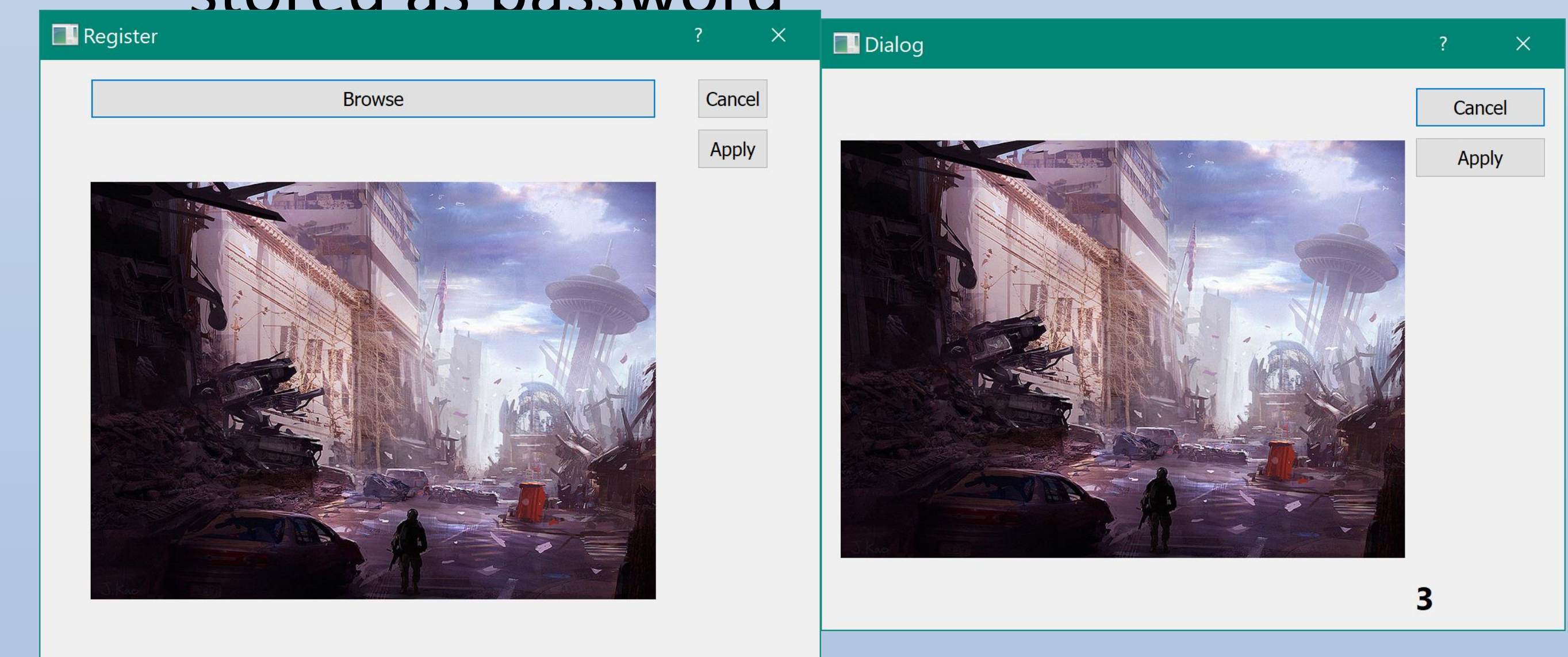
- Design and develop a C++ library that implements the algorithm described here
- The system should have functions to accept click point and generate a password hash, create GUI windows, compare hashes

Design and algorithm

- Developed by S. Chiasson et. al. [1]



- Line divided into segments based on location of click point gives unique segment identifier d
- Same approach in 2D yields two identifiers – $19^2 = 361$ possible grids [1]
- r is specified as a tolerance so approximate entries are accepted
- SHA-1 hash of grid identifier + (segment number in which click point is located) is stored as password



References:

1. S. Chiasson, J. Srinivasan, R. Biddle and P. C. V. Oorschot, "Centered Discretization with Application to Graphical Passwords," in Proceedings of USENIX UPSEC, 2008.
2. S. Chiasson, P. C. v. Oorschot and R. Biddle, "Graphical Password Authentication using Cued Click Points," in ESORICS, 2007.
3. N. Memon, "Graphical Passwords based on Robust Discretization," in IEEE Transactions on Information Forensics and Security, 2006.