## 1 Fuzzy Password Authenticated Key Exchange

```
\frac{\textbf{Alice: }PW_{Alice}, \textbf{ k}}{VK, SK \leftarrow ED25519.Gen()}
                                                                                                          Bob: PW_{Bob}, k
N = len(PW_{Alice})
\mathbf{for}\ i=1..N\ \mathbf{do}
                                                                                                          for i=1..N do
    IV \leftarrow \$ \left\{ 0,1 \right\}^{KeySize}
     x \leftarrow EC25519.GenPrivateKey()
                                                                                                            y \leftarrow EC25519.GenPrivateKey()
     X \leftarrow x.GenPublicKey(): g^x
                                                                                                            y \leftarrow Y.GenPublicKey(): g^y
     symmEncKey_A \leftarrow HKDFExpand(PW_{Alice}[i]||VK)
                                                                                                             symmEncKey_B \leftarrow HKDFExpand(PW_{Alice}[i]||\epsilon)
     X^* \leftarrow AES_{256}Enc(symmEncKey_A, X, IV)
                                                                                 X^*, VK, IV
                                                                                                            Y^* \leftarrow AES_{256}Enc(symmEncKey_B, Y, IV)
                                                                                                             symmDecKey_B \leftarrow HKDFExpand(PW_{Bob}[i]||VK)
                                                                                                             Z \leftarrow y.exchange(AES_{256}Dec(symmDecKey_B, X^*, IV)
                                                                                                             abort \mathbf{if} VK changes or VK not a Verification Key
        symmDecKey_A \leftarrow HKDFExpand(PW_{Alice}[i]||\epsilon)
        Z \leftarrow x.exchange(AES_{256}Dec(symmDecKey_A, Y^*, IV)
        K[i] \leftarrow SHA256(X^*, Y^*, Z)
                                                                                                            K[i] \leftarrow SHA256(X^*, Y^*, Z)
     endfor
                                                                                                       endfor
  C \leftarrow ReedSolomon.encode(U)
  E \leftarrow C - K
                                                                               \sigma_E, E, VK, EK
  \sigma_E \leftarrow Sign(SK, E)
                                                                                                          Abort \mathbf{if} VK differs from previous VK
                                                                                                          Verify(VK,\sigma_E,E)
                                                                                                          C \leftarrow E - K
                                                                                                          U \leftarrow ReedSolomon.decode(C)
     \mathbf{return}\ U
                                                                                                          \mathbf{return}\ U
```