Separating Symmetric and Asymmetric Password-Authenticated Key Exchange

12th Conference on Security and Cryptography for Networks

Julia Hesse, IBM Research – Zurich, Switzerland

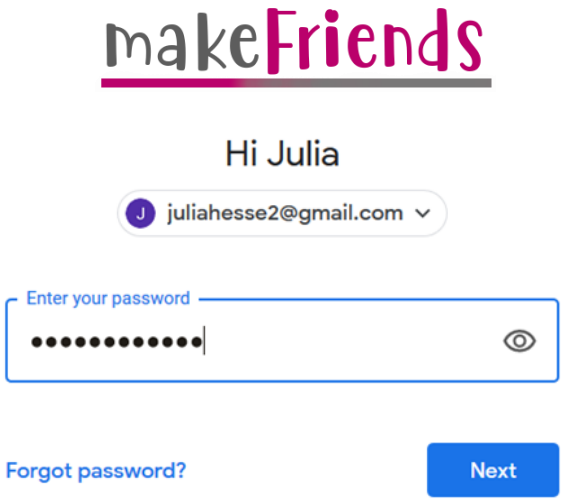
IBM

Introduction

to **a**symmetric **P**assword-**A**uthenticated **K**ey **E**xchange (**aPAKE**)

Have you ever wondered...

what happens when submitting your password via a login form?



Password-over-TLS

establish TLS connection

makeFriends

No cleartext passwords, only (salted) hashes!

Al!ce@makeF

verify password against Alice‘s account information

success/no success

Downside: Server learns Alice‘s password each time she logs in!

The formal setting

Abbey87	65aaf550277a586b2ecf159defe3ec51777605aff3cb
AliBaba	85db4f5e9c7515de5b27e9837449d7613e2b1a2689
Alice	**aacbd148f58b72f2012e8c0c1ac6ef953bb86f93fd7**
ArethaF	9686b7e5c0398fbcea5fa8a9a39b739ce10c16d2cd1
BarakO	a3241971a84f916a728b9f1a0bfc15b01247a799b1f

makeFriends

no leakage beyond outputs

password match/no match

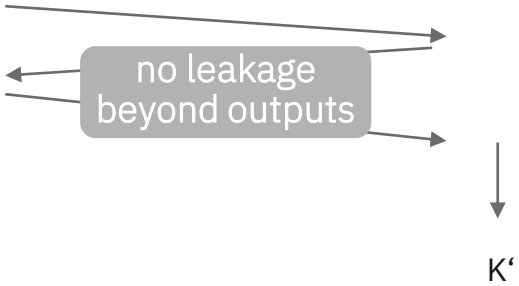
password match/no match

The cryptographic solution

Asymmetric Password-Authenticated Key Exchange (aPAKE)

Abbey87	65aaf550277a586b2ecf159defe3ec51777605aff3cb
AliBaba	85db4f5e9c7515de5b27e9837449d7613e2b1a2689
Alice	**aacbd148f58b72f2012e8c0c1ac6ef953bb86f93fd7**
ArethaF	9686b7e5c0398fbcea5fa8a9a39b739ce10c16d2cd1
BarakO	a3241971a84f916a728b9f1a0bfc15b01247a799b1f

makeFriends



K=K‘ if password was correct
otherwise K,K‘ random
(add Key Confirmation to learn outcome)

Security notions

for aPAKE

Careful with low entropy!

Tricky: An adversary can always guess Alice's password with **non-negligible probability**.

Solution: Demand that a password guess is the **best possible attack.**

Security properties of aPAKE, more details

- Server **honest**: online dictionary attack is best possible strategy
  - Alice (and everybody else) has exactly one password guess per run of protocol
- Server **compromised**: offline dictionary attack possible but costly
  - If adversary steals the password file, she still needs to, e.g., compute one hash per password guess
- In an aPAKE execution the (possibly malicious) server does not learn anything about the password used by Alice beyond match/no match
- Output keys are pseudorandom

Existing security notions for aPAKE

Benhamouda, Pointcheval (2013): Verifier-based PAKE

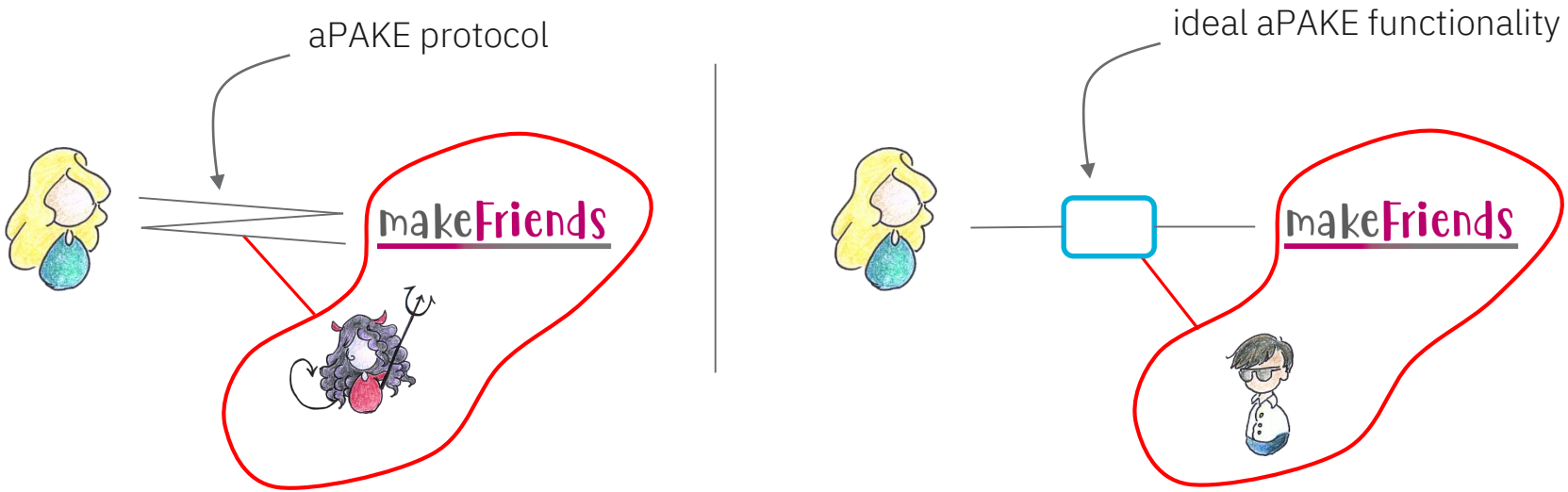
- game-based real-or-random notion
- multi-user setting
- passwords chosen at random for every user

Gentry, MacKenzie, Ramzan (2006): UC-secure aPAKE

- simulation-based notion (Universal Composability)
- single-user setting
- passwords adversarially chosen

focus of this talk.

Universal Composability (UC) in a nutshell

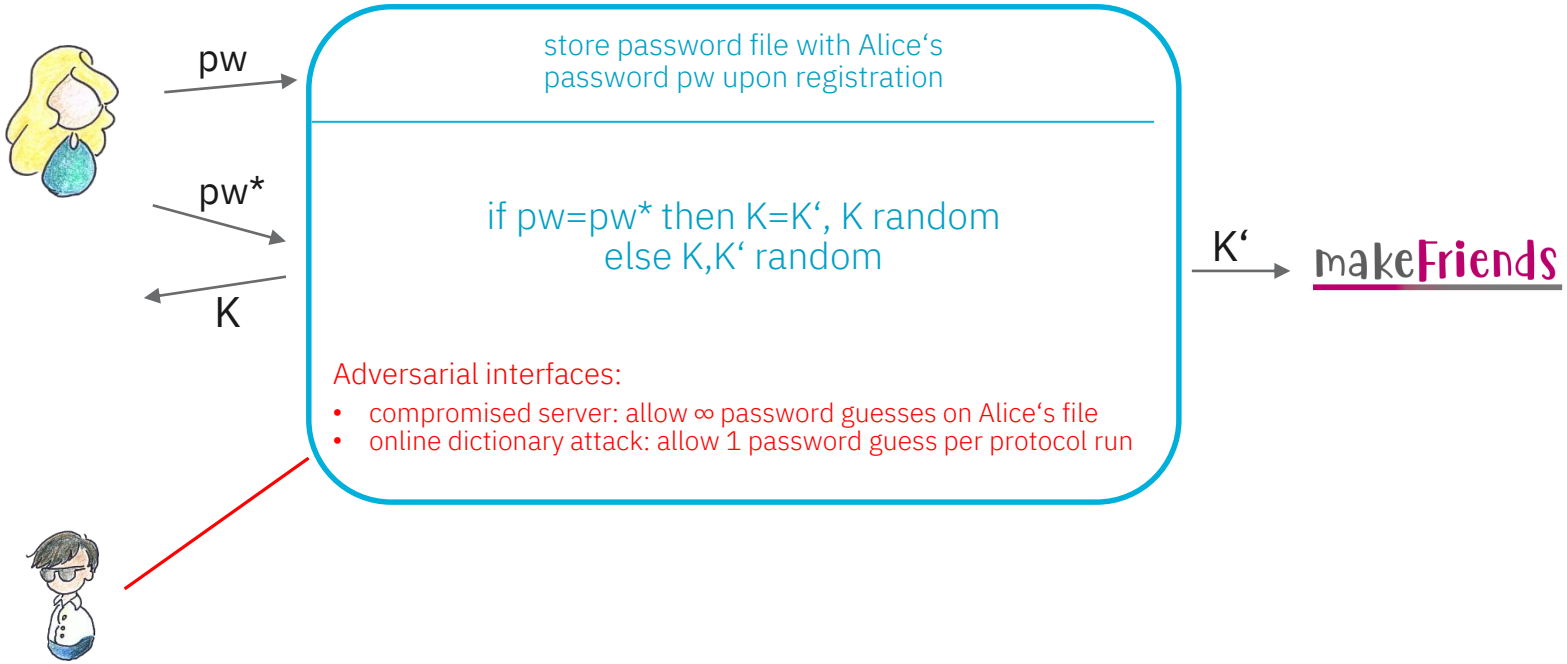


aPAKE protocol is secure if execution with adversary  looks like

ideal execution with simulator  .

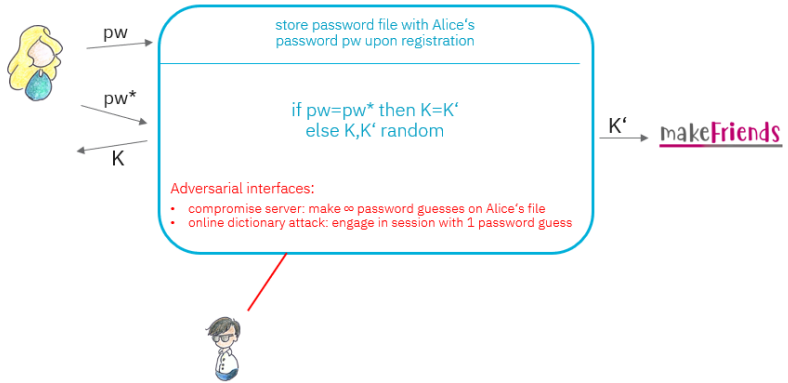
This holds even for a distinguisher choosing all passwords!

Ideal aPAKE functionality* (simplified)



* Craig Gentry, Philip MacKenzie, and Zulkar Ramzan. *A method for making password-based key exchange resilient to server compromise*. CRYPTO 2006.

UC-secure aPAKE



Notable constructions

- The Ω-protocol (C:GenMacRam06)
- OPAQUE (EC:JarKraXu18)
- AuCPace (CHES:HaaLab19)
- …

Main drawback:
All constructions use an idealized assumption such as a programmable random oracle (RO) or a generic group (GGM).

Disclaimer: UC secure *symmetric* PAKE (both parties use cleartext passwords) is possible in the standard model.

Our results

This paper

Question Is UC-secure asymmetric PAKE harder than symmetric PAKE?

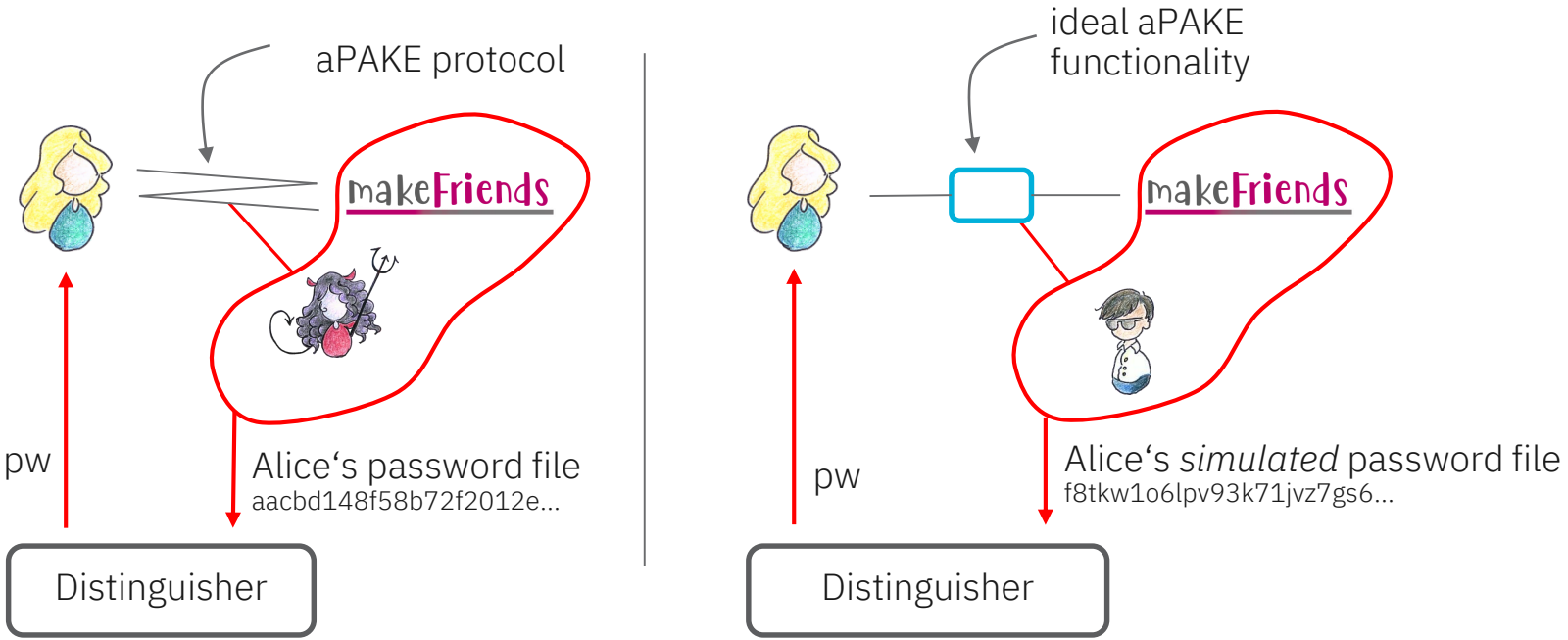
Answer Unfortunately, yes.

⟶ more evidence that idealized assumptions are indeed *required* by UC aPAKE.

Impossibility results

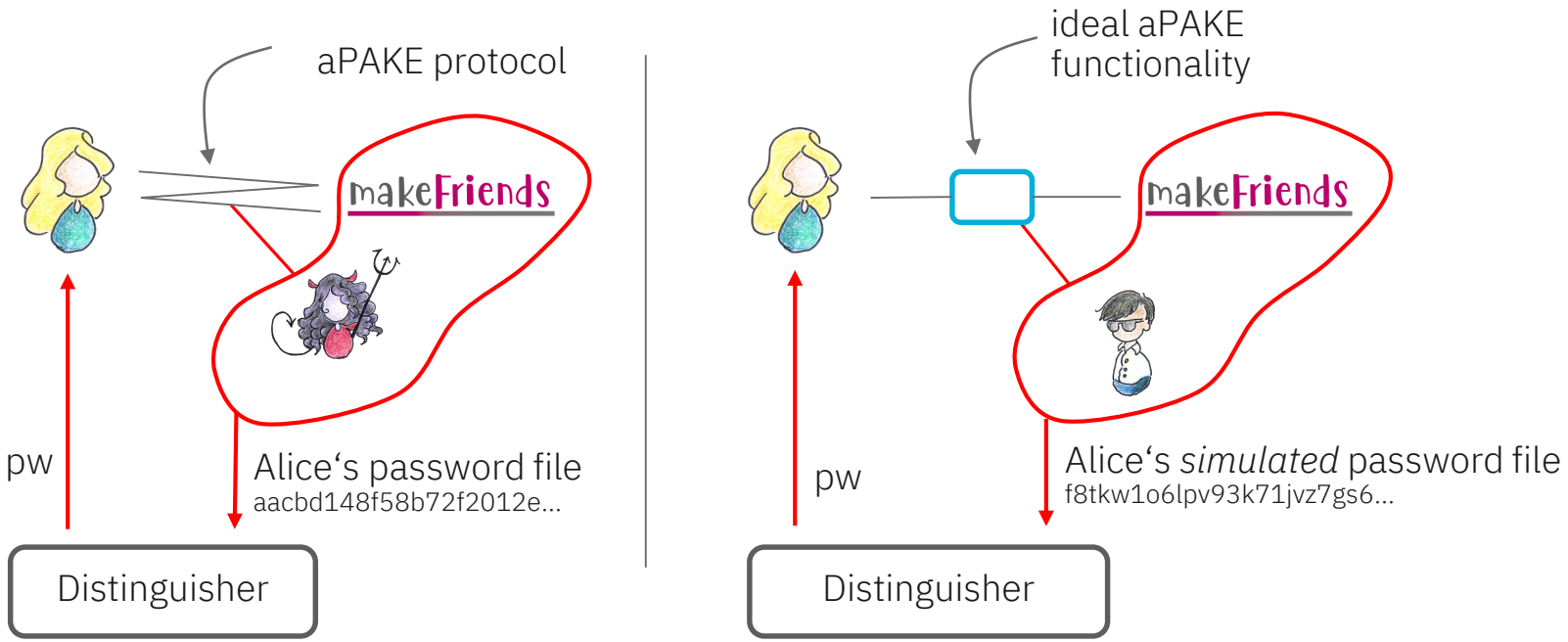
- UC aPAKE is impossible in the non-programmable RO model.
- It is also impossible in the programmable common reference string (CRS) model.

Commitment issues



Distinguisher checks whether pw in password file, e.g., whether SHA-3(pw) = aacbd148f58b72f2012e...

Commitment issues



Simulator needs to commit to password file *before* learning pw

- → *adaptive* programming required
- → neither non-programmable RO nor (prog. but non-adaptive) CRS enables UC secure aPAKE

Is UC secure aPAKE the wrong notion then?

No, please do analyze your aPAKE protocol in UC!

- It’s the only notion that considers adversarially chosen passwords
- It spares analysis of multi-user setting
- It captures complexity of brute-force attacks on password file properly
- Its modularity simplifies analysis of aPAKE integration into, e.g., TLS
- Proofs still meaningful: a successful attack *must* exploit weaknesses in instantiations of idealized assumption (e.g., the Hash function or the underlying group)

What's more - please have a look at the paper!

ia.cr/2019/1064

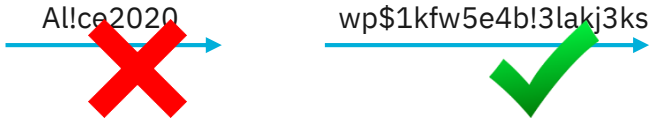
- Several fixes to UC aPAKE functionality*
- (First) full proof of security of an aPAKE called Ω-protocol*
- Implementation considerations for multi-session aPAKE

* Craig Gentry, Philip MacKenzie, and Zulkar Ramzan. *A method for making password-based key exchange resilient to server compromise*. CRYPTO 2006.

Take away slide

Asymmetric Password-Authenticated Key Exchange

- is a cool crypto primitive!  *
- spares you from sending your cleartext password!
- integrates nicely with TLS1.3!



*People drawn by Giorgia Azzurra Marson