In-band key-authentication from post-quantum key encapsulation mechanisms

Sebastian Verschoor
David R. Cheriton School of Computer Science
University of Waterloo
September 9th, 2021

Outline

Key authentication
Usability
Socialist Millionaire Protocol
Post-quantum solution
Intuition
Oblivious transfer
Private equality confirmation
Proof of security
Simple Universal Composability
Post-quantum security
Implementation
Discussion

Key authentication

- Secure messaging
  1. Trust establishment
  1.1 key exchange
  1.2 key authentication
  2. Conversation security
  3. Transport privacy
- Key authentication prevents Person-in-the-Middle attacks (and other impersonation attacks)

Certificates

- TLS uses certificates
- We want something without a trusted third party

Manual key fingerprint verification

Manual key fingerprint verification (cont.)
Key authentication: Usability

Usability issues lead to reduced security

- Studies where only 13% of users are able to successfully authenticate keys.

Observed problems with manual fingerprint comparison:

- Compare fingerprints in-band (note that the share button lets you do this).
- Compare only in one direction.
- Toggle “Mark as Verified” without actually verifying.

Observed user behaviour:

- Allowing in-band authentication increases usability.
- Users naturally rely on shared information.

Secret-based Zero-Knowledge verification

Implemented in OTR [AG07]

Two interfaces:

- Shared secret (mutual authentication).
- Question/Answer.

Pros:

- In-band.
- User sees no technical details (keys/fingerprints).

Cons:

- “Shared secrets require existing social relationships. This limits the usability of a system” [Ung+15].
- Synchronous.

No user study to confirming improved usability.

Socialist Millionaire Protocol

- Diffie-Hellman based protocol (not quantum-safe).
- Shares secrets vulnerable to harvest-and-decrypt.
- No direct translation to post-quantum primitives.
- Fairness abandoned in the OTR implementation.
- One user can abort after getting output.

Post-quantum solution

Proposed solution: KOP

- A (KEM-based Oblivious Transfer)-based Private Equality Confirmation.
A solution using envelopes [FNW96]
Binary inputs $x = x_1 x_2 \ldots x_s$ (Alice) and $y = y_1 y_2 \ldots y_s$ (Bob)
- Alice writes down $n$ random pairs $(A_i[0], A_i[1])$.
- Alice computes $a(x) = A_i[0] \oplus \cdots \oplus A_s[x_s]$
- Bob learns $a(y)$ as follows, Per pair:
  - Alice fills two envelopes, with $A_i[0]$ and $A_i[1]$
  - while Alice is not watching, Bob opens envelope $A_i[1]$
- $A_i[1] \oplus y$ is destroyed
- Switch roles, so Alice learns $b(x)$
- They compare $a(x) \oplus b(x)$ with $a(y) \oplus b(y)$

OT from KEMs

- Key encapsulation mechanism (KEM):
  - $(pk, sk) \leftarrow$ KeyGen()
  - $(k, ct) \leftarrow$ Encaps$(pk)$
  - $k \leftarrow$ Decaps$(sk, ct)$
- Public keys need to form a group $(G, +)$
- Decapsulation must not fail explicitly
  - Nor leak (implicit) failure through side-channel
- $m$ (local) random oracles $H_i : \mathbb{G}^{m-1} \rightarrow \mathbb{G}$

Output to both parties

The envelopes are only secure against semi-honest adversaries
- Simultaneous comparison (last step) is not possible
- Bob can reflect Alice’s last message to have her accept
- Use a cryptographic hash function $G$
- Alice sends $G(a(x)) \oplus b(x)$
- Bob rejects, or replies $a(y) \oplus b(y)$

Envelopes are realized by Oblivious Transfer (OT)
Endemic 1-out-of-$m$ OT ($m$ envelopes)
- If both Sender and Receiver are honest:
  - Receiver input $j$
  - Let $a[1], \ldots, a[m]$ be random values
  - Receiver gets output $a[j]$
  - Sender gets output $a[1], \ldots, a[m]$
- Malicious parties choose their own output
  - Malicious Sender sets $a[1], \ldots, a[m]$
  - Malicious Receiver sets $a[j]$

OT construction from KEMs [MR21]

Problem(?): Alice and/or Bob can send anything in the last message.
- A malicious party can force the other party to reject even when $x = y$
- Bob can even do this after having learned whether $x \neq y$
- In the context of key authentication this does not matter
- I call the resulting functionality Private Equality Confirmation (PEC)
Simple Universal Composability

Simple Universal Composability (SUC)

- Simulation paradigm (real/ideal)
- Environment $Z$
  - Wants to distinguish real model from ideal model
  - Chooses input and read outputs of parties $P_i$
  - Can corrupt parties
  - Interacts with the protocol (via the adversary interface)
- SUC-secure $\iff$ UC-secure
  - But SUC is less expressive than UC

Ideal model (functionality $F$)

- Dummy parties $P_i$
  - Non-corrupted parties only forward input/output
  - Private messages
- Simulator $S$
  - Controls input/output of corrupted parties

Hybrid model

- Simulator $S$
  - Goal: generate identically distributed view for $Z$
  - $S'$: defined relative to $A$
  - $Z$ is external to $S$: no rewinding
  - $S$ has to extract the effective input of the corrupted party to $F$
  - Can run code of honest parties itself
  - Can see output of corrupted parties
- Hard to prove anything in this plain model
  - Replace the real model with a hybrid model

SUC-security: For every adversary $A$ there must be a $S$ such that for all environments $Z$ on any advice $z$:

$$\Pr[suc\text{-real} = 1] - \Pr[suc\text{-ideal} = 1] = negl(\lambda)$$
Simple Universal Composability

Hybrid model: protocol \( \pi \) uses functionality \( F' \)

- SUC composition theorem:
  
  \[ \text{if } \pi \text{ is SUC-secure computes } F \text{ in the } F'-\text{hybrid model,} \]
  
  \[ \text{and } \rho \text{ securely computes } F' \text{ in the } F''-\text{hybrid model,} \]
  
  \[ \text{then } \sigma'' \text{ securely computes } F' \text{ in the } F''-\text{hybrid model} \]

  - \( \pi'' \) replace each invocation of \( F' \) by executing \( \rho \)
  - \( S \) usually runs \( F' \) in the simulation
  - Can see adversary input
  - Can choose output (distributed similarly)

- Rarely go all the way to real model
  - In this case: the random oracle model is the lowest hybrid

 Pact protocol

\[
\begin{align*}
m_A &= G(\alpha(x)) \oplus \beta(x) \\
\text{if } m_A &= \ldots \oplus \alpha(y) \oplus \beta(y) \quad \text{then: } m_A = \ldots \oplus \alpha(y) \oplus \beta(y) \\
\text{else: } m_B &= \text{reject}
\end{align*}
\]

S. R. Verschoor Key-authentication from KEMs 2021-09-09 26 / 37

SUC security of PEC

Hybrid argument to prove indistinguishability

- Start with a simulator that simply runs the honest party's code
  - trivially identical view for \( Z \)
  - invalid: requires knowledge of \( y \)
  - change it until it no longer requires \( y \) (but it will need \( F_{\text{prf}} \))
  - show each change is indistinguishable
  - Last hybrid is a valid simulator

PEC protocol

PEC protocol (simplified)

In this case: the random oracle model is the lowest hybrid

S. R. Verschoor Key-authentication from KEMs 2021-09-09 27 / 37

SUC security of PEC (corrupt Alice)
SUC security of PEC (corrupt Bob)

Two computational assumptions (in case $x \neq y$

- random $m_A$ should be indistinguishable from $G(\alpha(x)) \oplus \beta(x)$
- note that $\alpha(x)$ is uniformly random
- so this reduces to "G is pseudorandom"
- ideal model always rejects when $x \neq y$, real model might accept
- real Alice sends $m_A = G(\alpha(x)) \oplus \beta(x)$
- real Alice accepts $m_B = \alpha(x) \oplus \beta(x)$
- so this reduces to "G is one-way"

S. R. Verschoor Key-authentication from KEMs 2021–09–09 31 / 37

Post-quantum security

- Post-quantum security
  - Environment is a quantum machine (with quantum advice)
  - Assume a PQ-secure OT
  - Assume a PQ-secure $G$ (PQ one-way, PQ pseudorandom)
  - The security argument can be lifted to quantum security
    - No internal rewinding
    - Lifting does not necessarily preserve tightness
      - but the proof was asymptotic and non-uniform anyway

S. R. Verschoor Key-authentication from KEMs 2021–09–09 32 / 37

Implementation

libkop

- Hybrid KEM
  - Kyber (Round 3 CCA, NIST PQC lvl 5)
  - ECDH (Ed448 Goldilocks, Decaf)
    - with implicit failure on parsing error
  - C99 (~2000 LoC)
  - Side channel protection
    - Constant time
    - No secret indices
  - Domain separation ROMs

S. R. Verschoor Key-authentication from KEMs 2021–09–09 33 / 37

Performance

2-RTT protocol, 80-bit inputs ($m = 4, n = 40$)

- Message size
  - 254 KiB
  - 508 KiB
  - 254 KiB
  - 32 B
- Speed (ms)
  - 22
  - 114
  - 106
  - 15

S. R. Verschoor Key-authentication from KEMs 2021–09–09 34 / 37

Discussion

Key authentication from post-quantum KEMs (+ group structure)

Limitations

- OT security argument (despite claims) is not proven quantum-safe
- any Post-Quantum UC-secure OT suffices
- Asymptotic, non-uniform proof
- Rather heavy machinery
- Alternate solutions
  - Use alternative key authentication ceremony
  - Direct post-quantum replacement for SMP
  - PAKE

S. R. Verschoor Key-authentication from KEMs 2021–09–09 35 / 37
Thank you

References


References (cont.)


Socialist Millionaire Protocol

Quantum Lifting

▶ A simple hybrid argument [HSS11]:
For every adjacent hybrid $H_i, H_{i+1}$:
▶ there is a machine $M$ and classical distributions $D_i, D_{i+1}$
▶ so that $M(D_i) = H_i$ and $M(D_{i+1}) = H_{i+1}$
▶ and $D_i$ is quantum indistinguishable from $D_{i+1}$