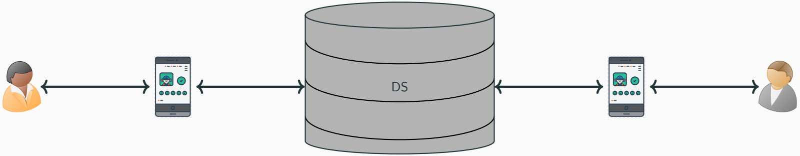
Secure Messaging Authentication against Active Man-in-the-Middle Attacks

---

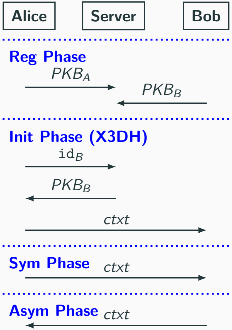
Benjamin Dowling ✉ *b.dowling@sheffield.ac.uk*
ETH Zurich, University of Sheffield
Britta Hale ✉ *britta.hale@nps.edu*
Naval Postgraduate School

2021-09-08

Signal as Asynchronous Protocol



Asynchronous messaging protocol:



Alice **Server** **Bob**

Reg Phase

PKB_A

PKB_B

Init Phase (X3DH)

$\mathtt{id}_B$

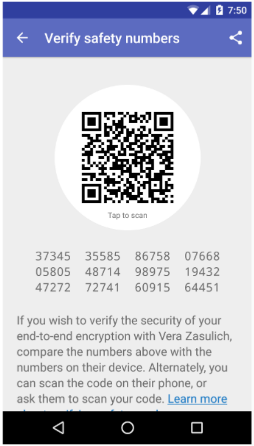
PKB_B

$ctxt$

Sym Phase $ctxt$

Asym Phase $ctxt$

Safety numbers!



37345 35585 86758 07668
05805 48714 98975 19432
47272 72741 60915 64451

If you wish to verify the security of your end-to-end encryption with Vera Zasulich, compare the numbers above with the numbers on their device. Alternately, you can scan the code on their phone, or ask them to scan your code. Learn more

Codes (QR & Numeric) contain long-term public keys and party identifiers:

$$\mathbf{local\_fprint} = H_i(0\|\texttt{fvers}\|idpk_A\|\texttt{id}_A, idpk_A)$$

$$\mathbf{remote\_fprint} = H_i(0\|\texttt{fvers}\|idpk_B\|\texttt{id}_B, idpk_B)$$

$$\textit{safety number} = \mathbf{local\_fprint}\|\mathbf{remote\_fprint}$$

Claim: this will *"verify the security of [the users] encryption."* (Signal App)

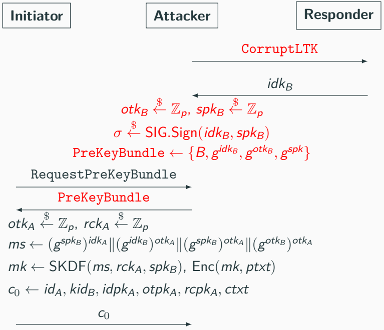
We identify weaknesses within this safety number authentication construct.

Issue 1: No Session Authentication

- Safety number is computed from purely public (and static) information.
  $\textbf{local\_fprint} =$
  $H_i(0 \| \texttt{fvers} \| idpk_A \| \texttt{id}_A, idpk_A)$
  $\textbf{remote\_fprint} =$
  $H_i(0 \| \texttt{fvers} \| idpk_B \| \texttt{id}_B, idpk_B)$

- If an attacker has learned the long-term secret key of the communicating partner, then creating a *PreKeyBundle* is trivial

- Impersonating attacks possible, verifying the Safety Number does not detect this attack



Initiator **Attacker** **Responder**

$\texttt{CorruptLTK}$

idk_B

$otk_B \xleftarrow{\$} \mathbb{Z}_p,\ spk_B \xleftarrow{\$} \mathbb{Z}_p$

$\sigma \xleftarrow{\$} \mathsf{SIG.Sign}(idk_B, spk_B)$

$\texttt{PreKeyBundle} \leftarrow \{B, g^{idk_B}, g^{otk_B}, g^{spk}\}$

$\texttt{RequestPreKeyBundle}$

$\texttt{PreKeyBundle}$

$otk_A \xleftarrow{\$} \mathbb{Z}_p,\ rck_A \xleftarrow{\$} \mathbb{Z}_p$

$ms \leftarrow (g^{spk_B})^{idk_A} \| (g^{idk_B})^{otk_A} \| (g^{spk_B})^{otk_A} \| (g^{otk_B})^{otk_A}$

$mk \leftarrow \mathsf{SKDF}(ms, rck_A, spk_B),\ \mathsf{Enc}(mk, ptxt)$

$c_0 \leftarrow id_A, kid_B, idpk_A, otpk_A, rcpk_A, ctxt$

c_0

User Authentication Phase

$\texttt{lf} = H_i(0 \| \texttt{f}_{\texttt{vers}} \| idpk_A \| id_A, idpk_A)$

$\texttt{rf} = H_i(0 \| \texttt{f}_{\texttt{vers}} \| idpk_B \| id_B, idpk_B)$

$\texttt{lf} = H_i(0 \| \texttt{f}_{\texttt{vers}} \| idpk_A \| id_A, idpk_A)$

$\texttt{rf} = H_i(0 \| \texttt{f}_{\texttt{vers}} \| idpk_B \| id_B, idpk_B)$

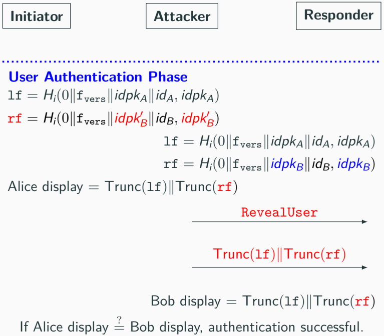
Alice display $= \mathsf{Trunc}(\texttt{lf}) \| \mathsf{Trunc}(\texttt{rf})$

Bob display $= \mathsf{Trunc}(\texttt{lf}) \| \mathsf{Trunc}(\texttt{rf})$

If Alice display $\overset{?}{=}$ Bob display, authentication successful.

Issue 2: Attacks possible with Display control

- Safety number is computed from purely public (and static) information. $\textbf{local\_fprint} = H_i(0\|\mathtt{fvers}\|idpk_A\|\mathtt{id}_A, idpk_A)$ $\textbf{remote\_fprint} = H_i(0\|\mathtt{fvers}\|idpk_B\|\mathtt{id}_B, idpk_B)$
- If an attacker can control user display via an overlay (access to secret state not necessary), then a forged safety number is displayed to the verifying party.



Initiator **Attacker** **Responder**

User Authentication Phase

$\mathtt{lf} = H_i(0\|\mathtt{f}_{\mathtt{vers}}\|idpk_A\|id_A, idpk_A)$

$\mathtt{rf} = H_i(0\|\mathtt{f}_{\mathtt{vers}}\|idpk'_B\|id_B, idpk'_B)$

$\mathtt{lf} = H_i(0\|\mathtt{f}_{\mathtt{vers}}\|idpk_A\|id_A, idpk_A)$

$\mathtt{rf} = H_i(0\|\mathtt{f}_{\mathtt{vers}}\|idpk_B\|id_B, idpk_B)$

Alice display $=$ Trunc($\mathtt{lf}$)$\|$Trunc($\mathtt{rf}$)

$\xrightarrow{\mathtt{RevealUser}}$

$\xrightarrow{\text{Trunc}(\mathtt{lf})\|\text{Trunc}(\mathtt{rf})}$

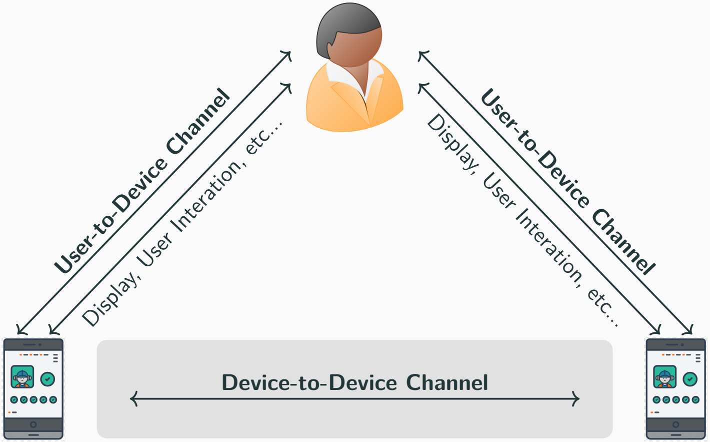
Bob display $=$ Trunc($\mathtt{lf}$)$\|$Trunc($\mathtt{rf}$)

If Alice display $\stackrel{?}{=}$ Bob display, authentication successful.

Contributions

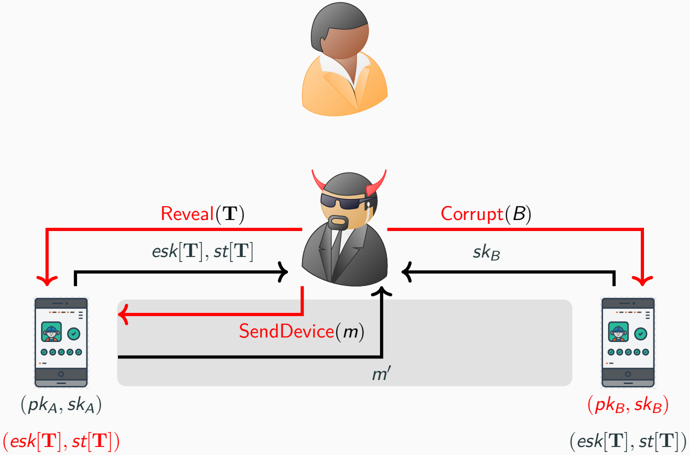
1. Security model to capture user-mediated authentication protocols (META)
2. Efficient and clean adaptation of Signal to achieve session authentication and per-epoch authentication: Modified Device-to-User Signal Authentication (MoDUSA)

META: Mediated Epoch Three-party Authentication Security Framework



High-level security goal: When a session at a Device "accepts", then there exists another honest and matching session (subset transcription matching).

META: Mediated Epoch Three-party Authentication Threat Model



$\mathsf{Reveal}(\mathbf{T})$

$esk[\mathbf{T}], st[\mathbf{T}]$

$\mathsf{Corrupt}(B)$

sk_B

$\mathsf{SendDevice}(m)$

m'

(pk_A, sk_A)

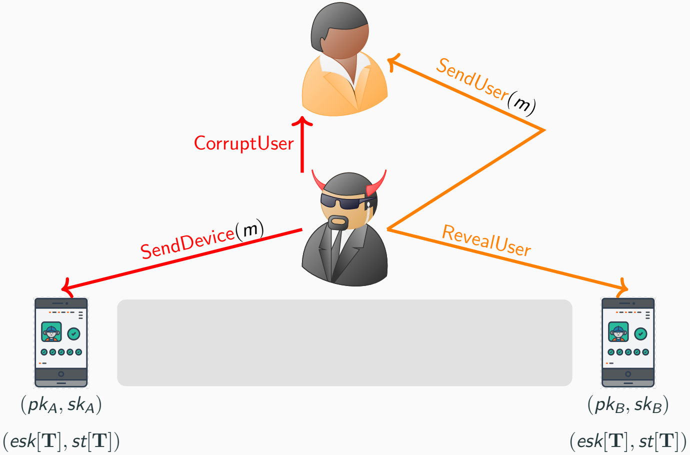
$(esk[\mathbf{T}], st[\mathbf{T}])$

(pk_B, sk_B)

$(esk[\mathbf{T}], st[\mathbf{T}])$

Attacker is capable of leaking long-term and device state.

META: Mediated Epoch Three-party Authentication Threat Model

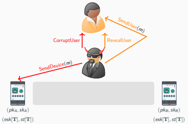


Attacker is able to compromise *directions* on the User-to-Device channel independently, sending messages between the User and the Devices.

Compromise Settings

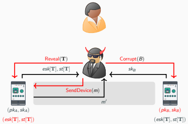
Compromised User:

- The attacker **cannot** leak current epoch secrets
- The attacker **cannot** RevealUser on **both** devices, allowing the attacker to inject messages from the Devices to the User.
- The attacker **cannot** CorruptUser, allowing the attacker to inject messages from the User to the Devices.

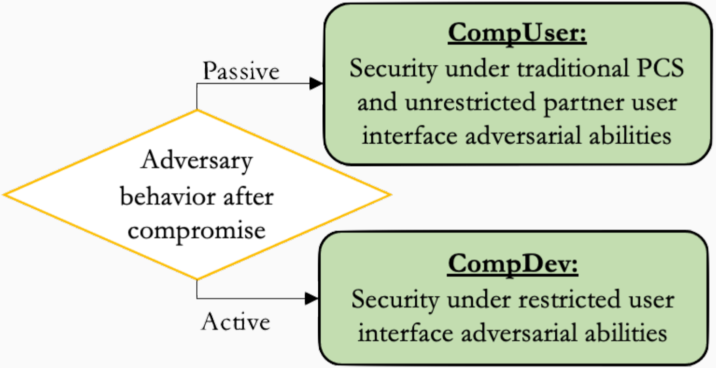


Compromised Device:

- The attacker **can** leak *any* secrets from the devices; and
- The attacker **cannot** RevealUser on **either** device, preventing the attacker from injecting any messages from the Devices to the User.
- The attacker **cannot** CorruptUser, allowing the attacker to inject messages from the User to the Devices.



Active Post-Compromise Security



CompUser:
Security under traditional PCS and unrestricted partner user interface adversarial abilities

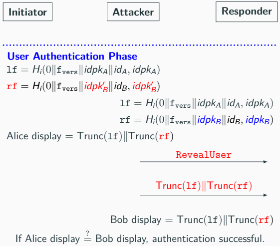
CompDev:
Security under restricted user interface adversarial abilities

Signal Authentication Insecure in Both Settings

Insecure Under Compromised User:

Tactic: Use RevealUser

Attack succeeds since Signal Safety Numbers are over purely public information.



Initiator **Attacker** **Responder**

User Authentication Phase

$\texttt{lf} = H_i(0\|\texttt{f}_{\texttt{vers}}\|idpk_A\|id_A, idpk_A)$

$\texttt{rf} = H_i(0\|\texttt{f}_{\texttt{vers}}\|idpk'_B\|id_B, idpk'_B)$

$\texttt{lf} = H_i(0\|\texttt{f}_{\texttt{vers}}\|idpk_A\|id_A, idpk_A)$

$\texttt{rf} = H_i(0\|\texttt{f}_{\texttt{vers}}\|idpk_B\|id_B, idpk_B)$

Alice display $= \mathsf{Trunc}(\texttt{lf})\|\mathsf{Trunc}(\texttt{rf})$

$\xrightarrow{\texttt{RevealUser}}$

$\xrightarrow{\mathsf{Trunc}(\texttt{lf})\|\mathsf{Trunc}(\texttt{rf})}$

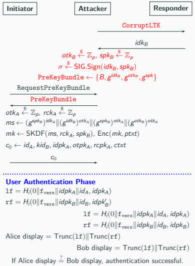
Bob display $= \mathsf{Trunc}(\texttt{lf})\|\mathsf{Trunc}(\texttt{rf})$

If Alice display $\stackrel{?}{=}$ Bob display, authentication successful.

Insecure under Compromised Device:

Tactic: Leaking session state, inject messages

Since Signal Safety Numbers doesn't authenticate per-session information, this attack is successful.



Initiator **Attacker** **Responder**

$\xrightarrow{\texttt{CorruptLTK}}$

$\xleftarrow{idk_B}$

$otk_B \xleftarrow{\$} \mathbb{Z}_p,\ spk_B \xleftarrow{\$} \mathbb{Z}_p$

$\sigma \xleftarrow{\$} \mathsf{SIG.Sign}(idk_B, spk_B)$

$\texttt{PreKeyBundle} \leftarrow \{B, g^{idk_B}, g^{otk_B}, g^{spk}\}$

$\xrightarrow{\texttt{RequestPreKeyBundle}}$

$\xleftarrow{\texttt{PreKeyBundle}}$

$otk_A \xleftarrow{\$} \mathbb{Z}_p,\ rck_A \xleftarrow{\$} \mathbb{Z}_p$

$ms \leftarrow (g^{spk_B})^{idk_A}\|(g^{idk_B})^{otk_A}\|(g^{spk_B})^{otk_A}\|(g^{otk_B})^{otk_A}$

$mk \leftarrow \mathsf{SKDF}(ms, rck_A, spk_B),\ \mathsf{Enc}(mk, ptxt)$

$c_0 \leftarrow id_A, kid_B, idpk_A, otpk_A, rcpk_A, ctxt$

$\xrightarrow{c_0}$

User Authentication Phase

$\texttt{lf} = H_i(0\|\texttt{f}_{\texttt{vers}}\|idpk_A\|id_A, idpk_A)$

$\texttt{rf} = H_i(0\|\texttt{f}_{\texttt{vers}}\|idpk_B\|id_B, idpk'_B)$

$\texttt{lf} = H_i(0\|\texttt{f}_{\texttt{vers}}\|idpk_A\|id_A, idpk_A)$

$\texttt{rf} = H_i(0\|\texttt{f}_{\texttt{vers}}\|idpk_B\|id_B, idpk_B)$

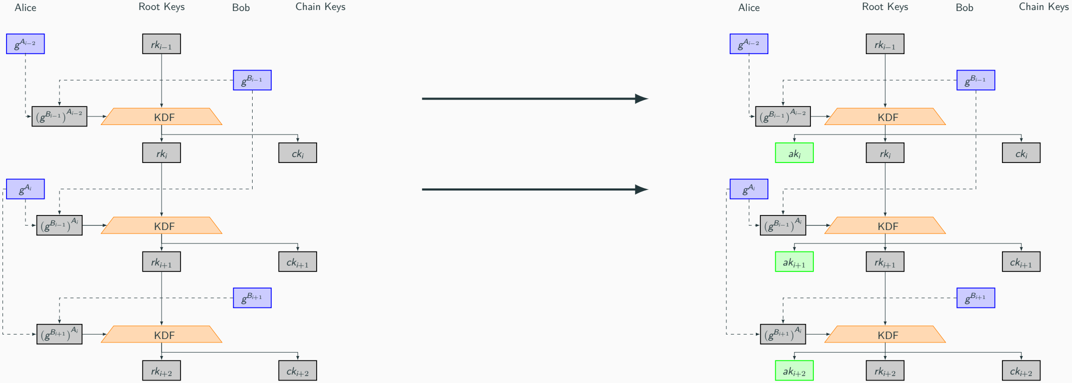
Alice display $= \mathsf{Trunc}(\texttt{lf})\|\mathsf{Trunc}(\texttt{rf})$

Bob display $= \mathsf{Trunc}(\texttt{lf})\|\mathsf{Trunc}(\texttt{rf})$

If Alice display $\stackrel{?}{=}$ Bob display, authentication successful.

MoDUSA: Modified Device-to-User Signal Authentication

Modify the Signal Key Schedule to add an additional authentication key.

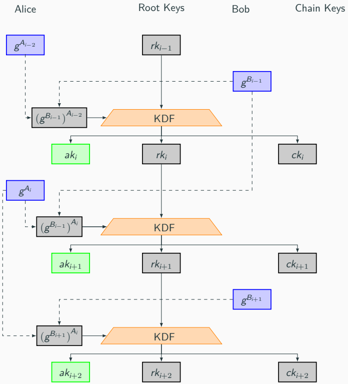


Hashed Transcripts

Pair of hashed transcripts of all public-key values sent between the two parties. Pair maintained in case (due to asynchronicity) one party has not received the most recent ratchet public key.

$$\boxed{H^{i-1}} = \text{Hash}\left(\boxed{PKB} \;\|\; \boxed{g^{A_0}} \;\|\; \boxed{g^{B_0}} \;\|\; \boxed{\ldots} \;\|\; \boxed{g^{A_i}} \right)$$

$$\boxed{H^{i}} = \text{Hash}\left(\boxed{PKB} \;\|\; \boxed{g^{A_0}} \;\|\; \boxed{g^{B_0}} \;\|\; \boxed{\ldots} \;\|\; \boxed{g^{A_i}} \;\|\; \boxed{g^{B_i}} \right)$$



New Safety Numbers

Safety numbers now update per epoch. Verification of safety number implies agreement on transcript of all cryptographic information.

$$\boxed{FP^{i-1}} = \mathrm{MAC}\left(\boxed{ak^{i-1}} , \boxed{H^{i-1}} \parallel \boxed{\text{role}} \right)$$

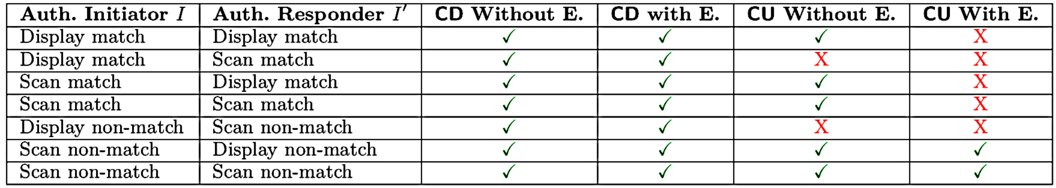
$$\boxed{FP^{i}} = \mathrm{MAC}\left(\boxed{ak^{i}} , \boxed{H^{i}} \parallel \boxed{\text{role}} \right)$$

Role separation prevents reflection attacks.

Maintain pair of safety numbers to account for potential asynchronicity.

Results of Analysing MoDUSA in META

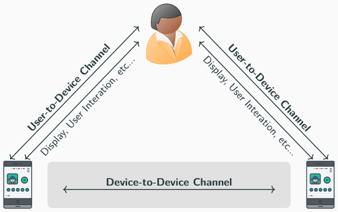
MoDUSA:

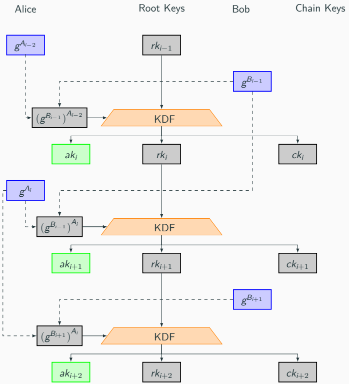


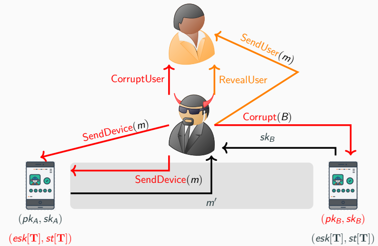
Auth. Initiator I	**Auth. Responder I'**	**CD Without E.**	**CD with E.**	**CU Without E.**	**CU With E.**
Display match	Display match	✓	✓	✓	X
Display match	Scan match	✓	✓	X	X
Scan match	Display match	✓	✓	✓	X
Scan match	Scan match	✓	✓	✓	X
Display non-match	Scan non-match	✓	✓	X	X
Scan non-match	Display non-match	✓	✓	✓	✓
Scan non-match	Scan non-match	✓	✓	✓	✓

- **CD**:**Compromised Device**.
- **CU**: **Compromised User**.
- **E**: Eavesdropper.

Summary


User-to-Device Channel
Display, User Interaction, etc...
User-to-Device Channel
Display, User Interaction, etc...
Device-to-Device Channel


Alice
Root Keys
Bob
Chain Keys
$g^{A_{i-2}}$
rk_{i-1}
$g^{B_{i-1}}$
$(g^{B_{i-1}})^{A_{i-2}}$
KDF
ak_i
rk_i
ck_i
g^{A_i}
$(g^{B_{i-1}})^{A_i}$
KDF
ak_{i+1}
rk_{i+1}
ck_{i+1}
$g^{B_{i+1}}$
$(g^{B_{i+1}})^{A_i}$
KDF
ak_{i+2}
rk_{i+2}
ck_{i+2}


SendUser(m)
CorruptUser
RevealUser
Corrupt(B)
SendDevice(m)
sk_B
SendDevice(m)
m'
(pk_A, sk_A)
$(esk[\mathbf{T}], st[\mathbf{T}])$
(pk_B, sk_B)
$(esk[\mathbf{T}], st[\mathbf{T}])$