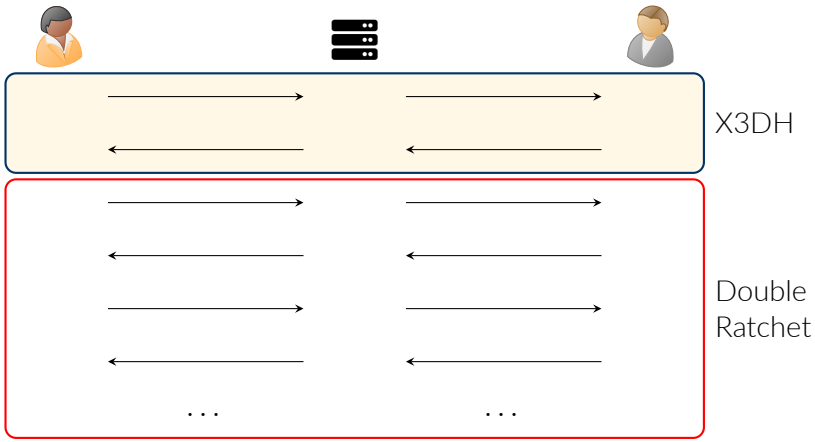
Post-Quantum X3DH

Thomas Prest (joint work with Keitaro Hashimoto,
Shu Katsumata and Kris Kwiatkowski)

PQShield

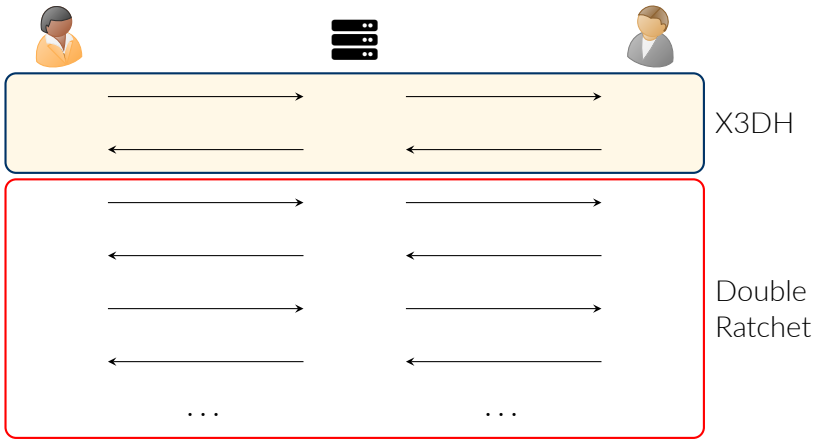
Secure messaging and the Signal protocol



Secure instant messaging:

- Notable differences with TLS: asynchrony, long-lived sessions, etc.
- The security gold standard is the Signal protocol (used in Signal, WhatsApp, etc.)
- Two main components: X3DH and Double Ratchet

Secure messaging and the Signal protocol



How about post-quantum/generic alternatives?

- *For Double Ratchet (generic):* [ACD19]
- *For X3DH (semi-generic):* this talk, based on [HKKP21]

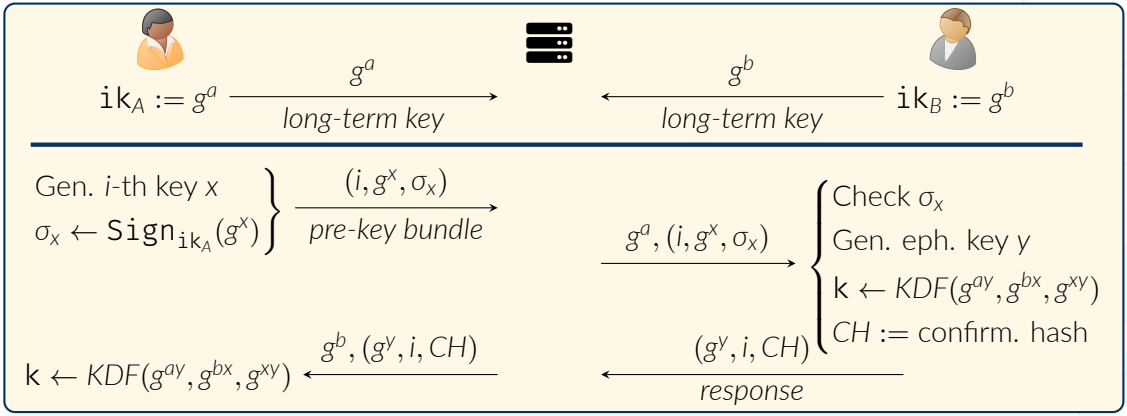
Roadmap

We focus on X3DH:

1. What does it do?
2. What properties do we expect from it?
3. How do we satisfy these properties in a PQ setting?
4. Can we provide a generic construction?

The end result is a PQ alternative (in various flavours) to X3DH.

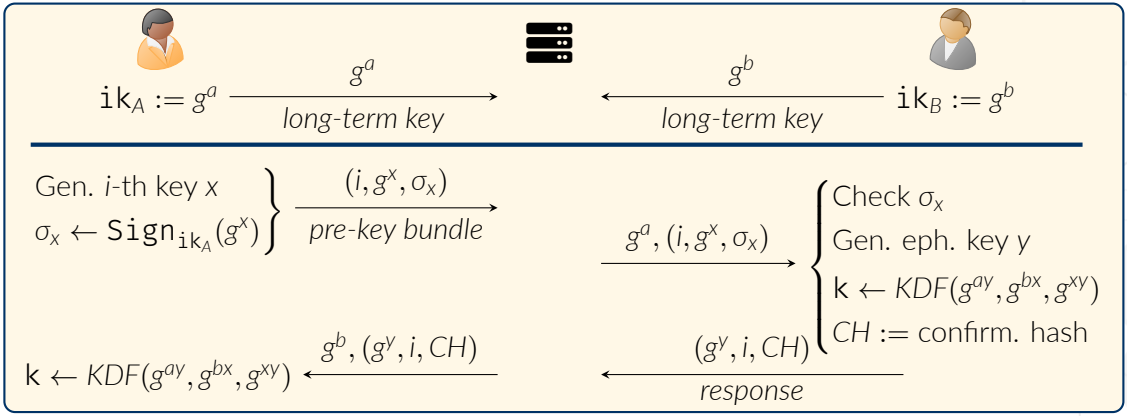
X3DH: reverse-engineering the spec [MP16]



Initial comments:

- Builds upon 3DH (same protocol without σ_x and CH).
- Note the two-message flow and the "*receiver-obliviousness*" of pre-key bundle.
- Parties delete ephemeral keys (x and y) as soon as they can (omitted in figure).

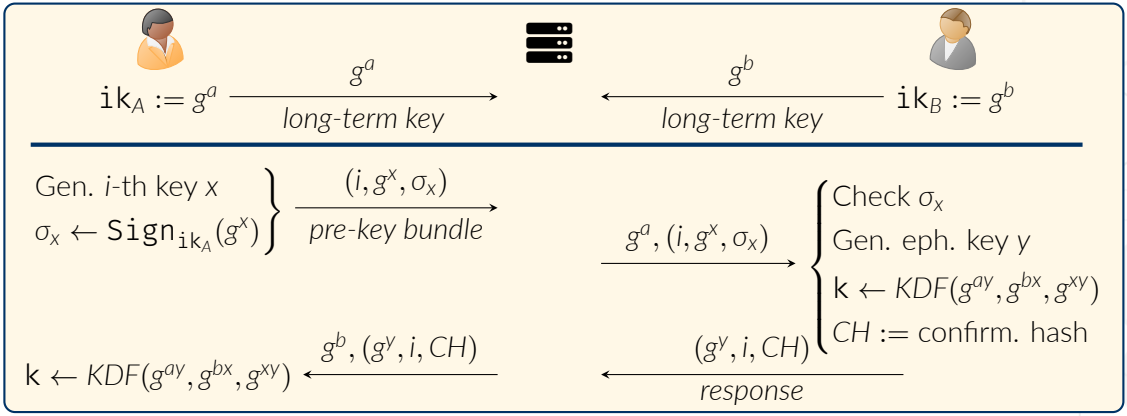
X3DH: reverse-engineering the spec [MP16]



CH is essentially a confirmation hash in disguise [MP16]:

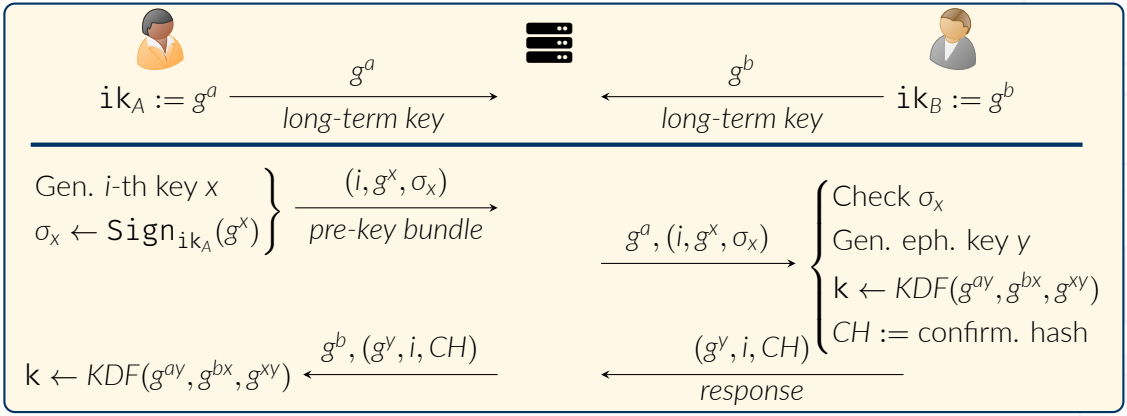
> " [CH is a] ciphertext encrypted with some AEAD encryption scheme using [AD = Encode($\mathtt{ik}_A$) || Encode($\mathtt{ik}_B$)] as associated data and using an encryption key which is either [k or derived from k]. "

X3DH: mutual authentication and forward secrecy



" Note that $[g^{ay}]$ and $[g^{bx}]$ provide mutual authentication, while $[g^{xy}$ provides] forward secrecy. "

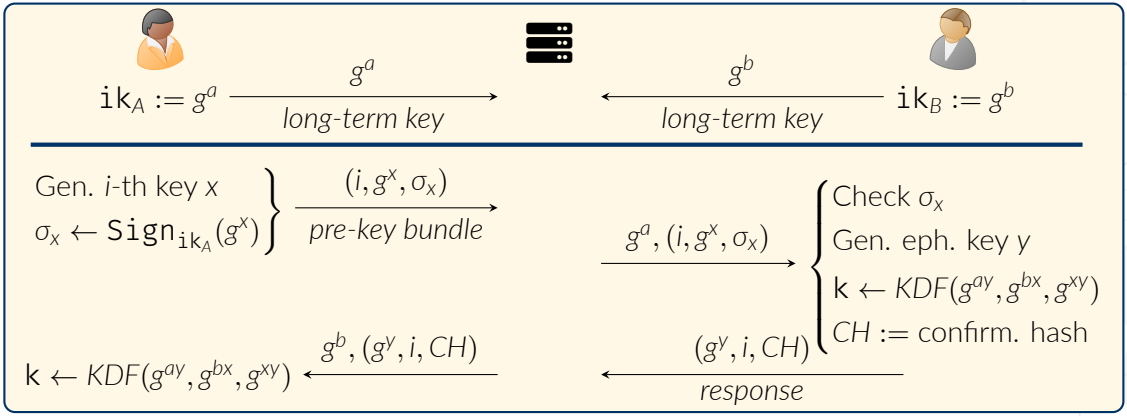
X3DH: mutual authentication and forward secrecy



A few comments on this topic:

- k is hidden as long as adversary doesn't learn $(a \wedge x) \vee (b \wedge y) \vee (x \wedge y)$.
- CH "upgrades" implicit authentication of Bob to explicit.

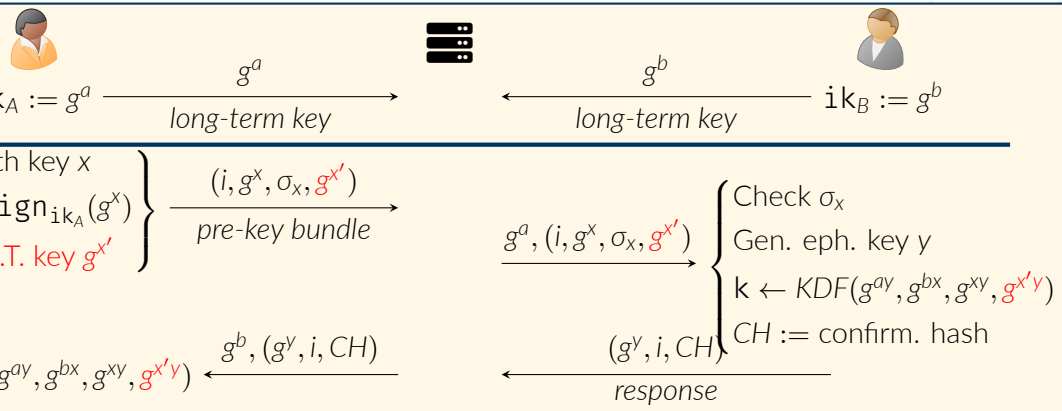
X3DH: deniability



> " X3DH doesn't give either Alice or Bob a publishable cryptographic proof of the contents of their communication or the fact that they communicated. "

- X3DH may only achieve *offline* (as opposed to *online*) deniability [MP16].
- Adversarial model: *semi-honest* (*malicious* under stronger assumptions [VGIK20]).

X3DH: one-time keys (optional)



X3DH allows to provide an optional one-time key.

- *Motivation:* forward secrecy and protection against replay attacks
- *Replay attacks:* there are simpler ways to thwart them
- *Forward secrecy:* largely addressed by signed pre-keys and Double Ratchet

List of desirable properties

Functional properties:

- *Two-message flow:* Initiator → Responder → Initiator
- *Receiver-obliviousness:* first message is independent of the Responder[1]

Security properties:

- *Confidentiality:* session keys looks pseudorandom except for intended parties
  - Weak perfect forward secrecy (wPFS)
  - Perfect forward secrecy (PFS)
- *Mutual authentication:* Initiator knows she talks to Responder, and reciprocally
  - Includes *Key-Compromise Impersonation* (KCI) attacks[2]
  - Includes *Unknown Key-Share* (UKS) attacks[3]
- *Deniability:* ~~online/~~offline, against semi-honest/malicious adversaries

[1] Also known as *post-specified peers*.

[2] KCI: The adversary uses A's long-term key to impersonate other parties towards A.

[3] UKS: A and B compute the same key, but A believe he's talking to C.

How hard is it to devise a PQ alternative to X3DH?

PQ SHIELD

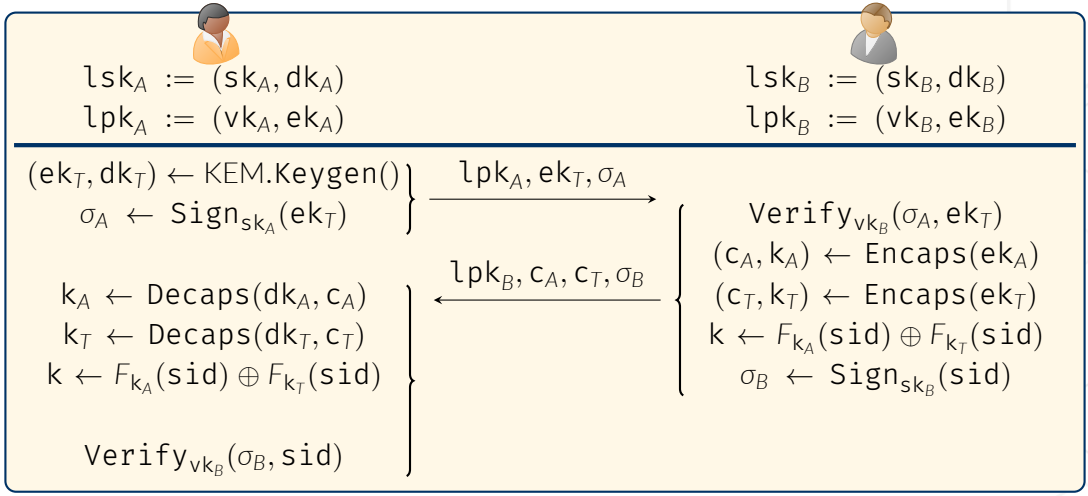
Natural approach #1: replace classical Diffie-Hellman by post-quantum variant.

1. *Noisy Diffie-Hellman* (lattices and codes) [DXL12, Pei14, ADPS16]
   - Doesn't support static keys [Flu16, DFR18]
2. Gap variant of CSIDH (isogenies) [dKsV20, KTAT20, BFG+20]
   - Slow
   - Quantum security of *standard* CSIDH is debated [BLMP19, BS20, Pei20, CCJR20]
3. SIDH and variants thereof (isogenies) [DG21]
   - Uses a SIDH proof of knowledge [FDGZ21] to thwart adaptive attacks [GPST16].
   - The size of the SIDH proof of knowledge needs to be determined.

Natural approach #2: build a protocol that captures as many properties of X3DH as possible using only generic building blocks.

- → KEMs
- → Signatures
- → PRFs
- → etc.

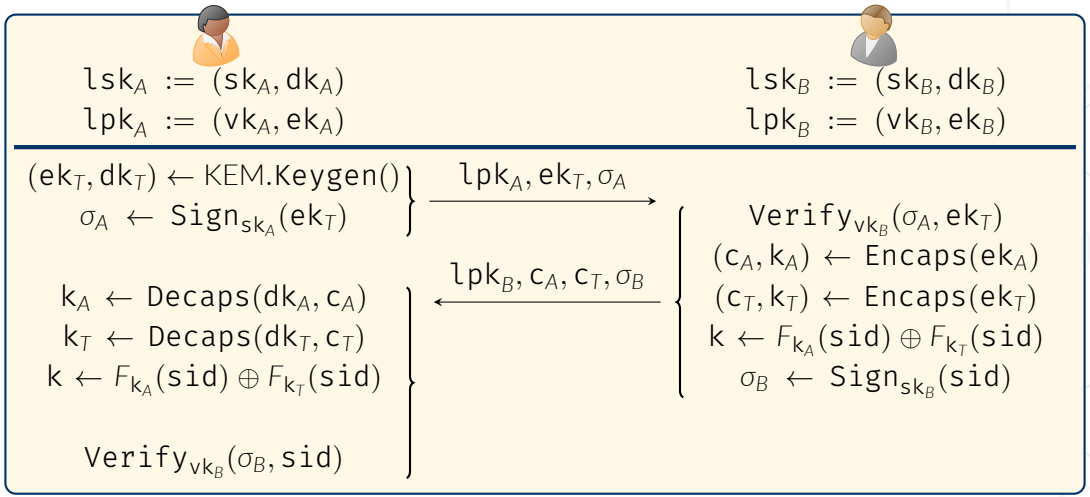
A first generic construction (V0)



Initial comments:

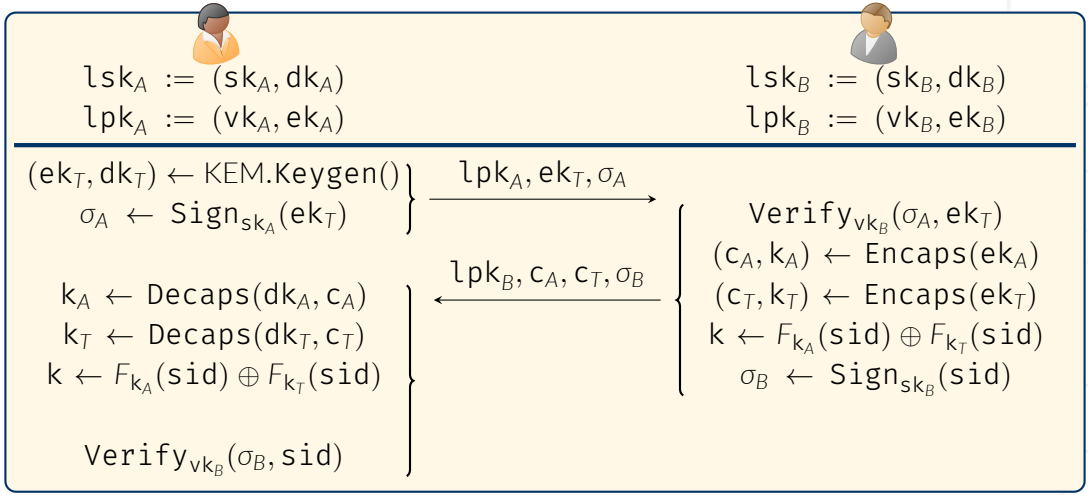
- $\mathtt{sid} = (\mathtt{lpk}_A \| \mathtt{lpk}_B \| \mathtt{ek}_A \| \mathtt{c}_A \| \mathtt{c}_T)$ is the session identifier.
- Omitted for clarity: $\mathtt{k}_A$ and $\mathtt{k}_T$ may be processed by randomness extractors.

A first generic construction (V0)



- *Leakage of long-term keys:* secrecy of $\mathtt{k}$ is guaranteed by $\mathtt{ek}_T$ and $\mathtt{c}_T$
- *State leakage (irrelevant for)*: secrecy of $\mathtt{k}$ is guaranteed by $\mathtt{c}_A$
- *KCI:* prevented by σ_B
- *UKS:* prevented by $\mathtt{sid}$ and $\mathtt{ek}_A$

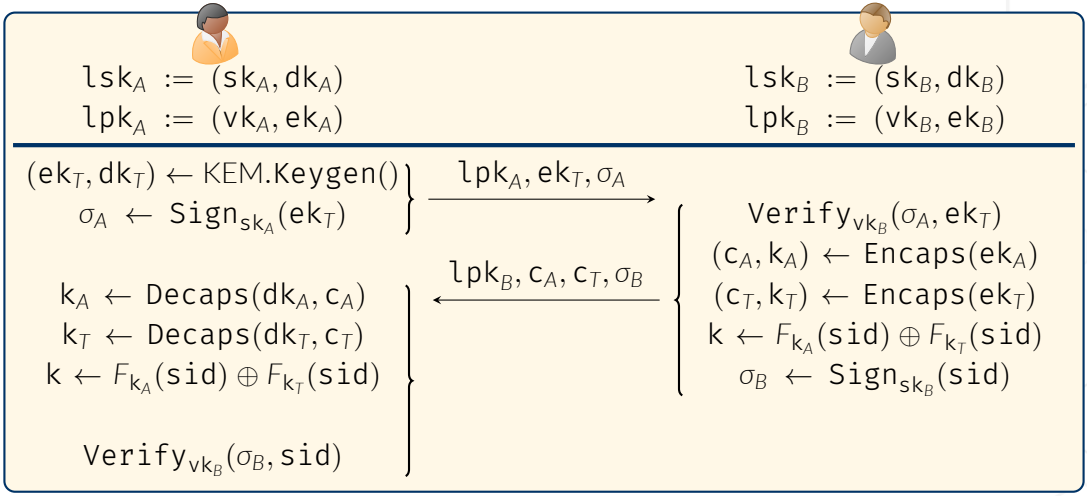
A first generic construction (V0)



→ is explicitly authenticated by his signature.

→ is implicitly authenticated since no one else than her may recover $\mathtt{k}$.

A first generic construction (V0)



Minor possible tweaks:

- *Omit σ_A:* Downgrades PFS to wPFS (note: Double Ratchet then provides PFS again)
- *NAXOS trick:* mitigates leakage of 's randomness by combining it with $\mathtt{lsk}_B$

A very weakly deniable version (V1)

$\mathtt{lsk}_A := (\mathtt{sk}_A, \mathtt{dk}_A)$
$\mathtt{lpk}_A := (\mathtt{vk}_A, \mathtt{ek}_A)$

$\mathtt{lsk}_B := (\mathtt{sk}_B, \mathtt{dk}_B)$
$\mathtt{lpk}_B := (\mathtt{vk}_B, \mathtt{ek}_B)$

$(\mathtt{ek}_T, \mathtt{dk}_T) \leftarrow \mathtt{KEM.Keygen}()$
$\sigma_A \leftarrow \mathtt{Sign}_{\mathtt{sk}_A}(\mathtt{ek}_T)$

$\xrightarrow{\mathtt{lpk}_A, \mathtt{ek}_T, \sigma_A}$

$\mathtt{Verify}_{\mathtt{vk}_B}(\sigma_A, \mathtt{ek}_T)$
$(\mathtt{c}_A, \mathtt{k}_A) \leftarrow \mathtt{Encaps}(\mathtt{ek}_A)$
$(\mathtt{c}_T, \mathtt{k}_T) \leftarrow \mathtt{Encaps}(\mathtt{ek}_T)$
$\mathtt{k} \| \mathtt{k}_\sigma \leftarrow F_{\mathtt{k}_A}(\mathtt{sid}) \oplus F_{\mathtt{k}_T}(\mathtt{sid})$
$\sigma_B \leftarrow \mathtt{Sign}_{\mathtt{sk}_B}(\mathtt{sid})$
$\mathtt{c} \leftarrow \sigma_B \oplus \mathtt{k}_\sigma$

$\xleftarrow{\mathtt{lpk}_B, \mathtt{c}_A, \mathtt{c}_T, \mathtt{c}}$

$\mathtt{k}_A \leftarrow \mathtt{Decaps}(\mathtt{dk}_A, \mathtt{c}_A)$
$\mathtt{k}_T \leftarrow \mathtt{Decaps}(\mathtt{dk}_T, \mathtt{c}_T)$
$\mathtt{k} \| \mathtt{k}_\sigma \leftarrow F_{\mathtt{k}_A}(\mathtt{sid}) \oplus F_{\mathtt{k}_T}(\mathtt{sid})$
$\sigma_B \leftarrow \mathtt{c} \oplus \mathtt{k}_\sigma$
$\mathtt{Verify}_{\mathtt{vk}_B}(\sigma_B, \mathtt{sid})$

We can tweak the protocol to achieve a weak flavour of deniability for free.

- We mask σ_B using a pseudorandom keystream derived from $\mathtt{k}_A, \mathtt{k}_T$ (tangerine).
- The transcript makes  anonymous as long as doesn't cooperate.
- However if leaks σ_B, then is bound.

A weakly deniable version (V1.5)

$\mathtt{lsk}_A := (\mathtt{sk}_A, \mathtt{dk}_A)$
$\mathtt{lpk}_A := (\mathtt{vk}_A, \mathtt{ek}_A)$

$\mathtt{lsk}_B := (\mathtt{sk}_B, \mathtt{dk}_B)$
$\mathtt{lpk}_B := (\mathtt{vk}_B, \mathtt{ek}_B)$

$(\mathtt{ek}_T, \mathtt{dk}_T) \leftarrow \mathtt{KEM.Keygen}()$
$\sigma_A \leftarrow \mathtt{Sign}_{\mathtt{sk}_A}(\mathtt{ek}_T)$

$\xrightarrow{\mathtt{lpk}_A, \mathtt{ek}_T, \sigma_A}$

$\mathtt{Verify}_{\mathtt{vk}_B}(\sigma_A, \mathtt{ek}_T)$
$(\mathtt{c}_A, \mathtt{k}_A) \leftarrow \mathtt{Encaps}(\mathtt{ek}_A)$
$(\mathtt{c}_T, \mathtt{k}_T) \leftarrow \mathtt{Encaps}(\mathtt{ek}_T)$
$\mathtt{k} \| \mathtt{k}_\sigma \leftarrow F_{\mathtt{k}_A}(\mathtt{sid}) \oplus F_{\mathtt{k}_T}(\mathtt{sid})$
$\sigma_B \leftarrow \mathtt{RS.Sign}_{\mathtt{sk}_B, \mathtt{ring}}(\mathtt{sid})$
$\mathtt{c} \leftarrow \sigma_B \oplus \mathtt{k}_\sigma$

$\xleftarrow{\mathtt{lpk}_B, \mathtt{c}_A, \mathtt{c}_T, c}$

$\mathtt{k}_A \leftarrow \mathtt{Decaps}(\mathtt{dk}_A, \mathtt{c}_A)$
$\mathtt{k}_T \leftarrow \mathtt{Decaps}(\mathtt{dk}_T, \mathtt{c}_T)$
$\mathtt{k} \| \mathtt{k}_\sigma \leftarrow F_{\mathtt{k}_A}(\mathtt{sid}) \oplus F_{\mathtt{k}_T}(\mathtt{sid})$
$\sigma_B \leftarrow \mathtt{c} \oplus \mathtt{k}_\sigma$
$\mathtt{RS.Verify}_{\mathtt{ring}}(\sigma_B, \mathtt{sid})$

- Main idea: replace signatures by *ring* signatures (pine green).
- Setting $\mathtt{ring} = \{\mathtt{vk}_A, \mathtt{vk}_B\}$ only gives a weak flavour of deniability.

A deniable version (offline, semi-honest) (V2)

$\mathtt{lsk}_A := (\mathtt{sk}_A, \mathtt{dk}_A)$
$\mathtt{lpk}_A := (\mathtt{vk}_A, \mathtt{ek}_A)$

$\mathtt{lsk}_B := (\mathtt{sk}_B, \mathtt{dk}_B)$
$\mathtt{lpk}_B := (\mathtt{vk}_B, \mathtt{ek}_B)$

$(\mathtt{ek}_T, \mathtt{dk}_T) \leftarrow \mathsf{KEM.Keygen}()$
$\sigma_A \leftarrow \mathtt{Sign}_{\mathtt{sk}_A}(\mathtt{ek}_T)$
$(\mathtt{sk}_T, \mathtt{vk}_T) \leftarrow \mathsf{RS.Keygen}()$

$\xrightarrow{\mathtt{lpk}_A, \mathtt{ek}_T, \sigma_A, \mathtt{vk}_T}$

$\mathtt{Verify}_{\mathtt{vk}_B}(\sigma_A, \mathtt{ek}_T)$
$(\mathtt{c}_A, \mathtt{k}_A) \leftarrow \mathtt{Encaps}(\mathtt{ek}_A)$
$(\mathtt{c}_T, \mathtt{k}_T) \leftarrow \mathtt{Encaps}(\mathtt{ek}_T)$
$\mathtt{k} \| \mathtt{k}_\sigma \leftarrow F_{\mathtt{k}_A}(\mathtt{sid}) \oplus F_{\mathtt{k}_T}(\mathtt{sid})$
$\sigma_B \leftarrow \mathsf{RS.Sign}_{\mathtt{sk}_B, \mathtt{ring}}(\mathtt{sid})$
$\mathtt{c} \leftarrow \sigma_B \oplus \mathtt{k}_\sigma$

$\xleftarrow{\mathtt{lpk}_B, \mathtt{c}_A, \mathtt{c}_T, \mathtt{c}}$

$\mathtt{k}_A \leftarrow \mathtt{Decaps}(\mathtt{dk}_A, \mathtt{c}_A)$
$\mathtt{k}_T \leftarrow \mathtt{Decaps}(\mathtt{dk}_T, \mathtt{c}_T)$
$\mathtt{k} \| \mathtt{k}_\sigma \leftarrow F_{\mathtt{k}_A}(\mathtt{sid}) \oplus F_{\mathtt{k}_T}(\mathtt{sid})$
$\sigma_B \leftarrow \mathtt{c} \oplus \mathtt{k}_\sigma$
$\mathsf{RS.Verify}_{\mathtt{ring}}(\sigma_B, \mathtt{sid})$

- Main idea: replace signatures by *ring* signatures (pine green).
- ~~Setting $\mathtt{ring} = \{\mathtt{vk}_A, \mathtt{vk}_B\}$ only gives a weak flavour of deniability.~~
- Setting $\mathtt{ring} = \{\mathtt{vk}_T, \mathtt{vk}_B\}$ with $\mathtt{vk}_T$ an ephemeral key makes (V2) truly deniable.

What about malicious adversaries?

Our protocols (V0-V2) are not deniable against *malicious* adversaries.
Example with the Raptor ring signature sheme:

Raptor [LAZ19] (≈ [RST01] + Falcon [PFH+17])

- Each public keys is a polynomial $\mathtt{vk}_i \in \mathbb{Z}_q[x]/(\varphi)$.
- A ring signature of $\mathtt{msg}$ for the ring $(\mathtt{vk}_i)_{i\in\mathcal{R}}$ consists of:
  - a salt $\mathtt{salt}$,
  - a short tuple $s_0 \| (s_i)_{i\in\mathcal{R}}$ such that:

$$s_0 + \sum_{i\in\mathcal{R}} s_i \cdot \mathtt{vk}_i = H(\mathtt{msg}, \mathtt{salt}) \qquad (\bigstar)$$

 can break  's deniability by setting her ephemeral ring signing key maliciously:

1. If $(\mathtt{vk}_T = 0)$, then $(\bigstar)$ becomes: $s_0 + s_B \cdot \mathtt{vk}_B = H(\mathtt{msg}, \mathtt{salt})$
2. If $(\mathtt{vk}_T = x^k)$, then $(\bigstar)$ becomes: $(s_0 + x^k \cdot s_T) + s_B \cdot \mathtt{vk}_B = H(\mathtt{msg}, \mathtt{salt})$

What about malicious adversaries?

Our protocols (V0-V2) are not deniable against *malicious* adversaries.
Example with the Raptor ring signature sheme:

Raptor [LAZ19] ($\approx$ [RST01] + Falcon [PFH+17])

- Each public keys is a polynomial $\mathtt{vk}_i \in \mathbb{Z}_q[x]/(\varphi)$.
- A ring signature of $\mathtt{msg}$ for the ring $(\mathtt{vk}_i)_{i\in\mathcal{R}}$ consists of:
  - a salt $\mathtt{salt}$,
  - a short tuple $s_0 \| (s_i)_{i\in\mathcal{R}}$ such that:

$$s_0 + \sum_{i\in\mathcal{R}} s_i \cdot \mathtt{vk}_i = H(\mathtt{msg}, \mathtt{salt}) \qquad (\bigstar)$$

 can break  's deniability by setting her ephemeral ring signing key maliciously:

1. If $(\mathtt{vk}_T = 0)$, then $(\bigstar)$ becomes: $s_0 + s_B \cdot \mathtt{vk}_B = H(\mathtt{msg}, \mathtt{salt})$
2. If $(\mathtt{vk}_T = x^k)$, then $(\bigstar)$ becomes: $(s_0 + x^k \cdot s_T) + s_B \cdot \mathtt{vk}_B = H(\mathtt{msg}, \mathtt{salt})$

See our paper for a more theoretical argument likely to encompass more (lattice-based) ring signature schemes.

A few more comments about deniability

Signature sizes for 2-users ring signatures vs standard signatures:

- *NTRU-based:* 2.5 KiB [LAZ19] vs 0.67 KiB [PFH+20]
- *M-LWE/SIS-based:* 4.4 KiB [YEL+21] vs 2.3 KiB [LDK+20]
- *CSIDH-based:* 3.5 KiB [BKP20] vs 0.26 KiB [BKV19]

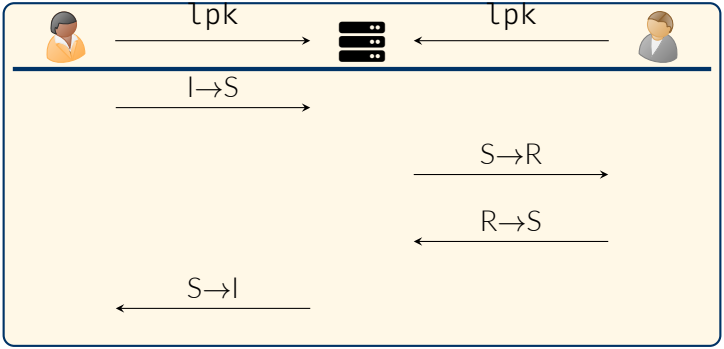
We can upgrade (V2) to a construction (V3) deniable against malicious adversaries:

- *Solution:* 🤦 sends a NIZK proof that she generated vk_T honestly
- *Open question:* can we do that efficiently?

A recent paper [BFG+21] has a construction similar to (V2), but replaces ring signatures with designated-verifier signatures (DVS).

- [BFG+21] shows (Ring signatures $\Rightarrow$ DVS).
- We show that (DVS $\Rightarrow$ Ring signatures), so both are equivalent.

Instantiation of (V1) at NIST level I (≈ AES-128)



Scheme	CPU cycles	lpk	I→S	S→R	R→S	S→I	Total
Falcon/Saber	4.2 MC	1569 B	1362 B	2931 B	2162 B	3731 B	10186 B
Falcon/SIKE	2614 MC	1227 B	1020 B	2247 B	1382 B	2609 B	7258 B
Dilithium/NTRU	6.9 MC	2011 B	3119 B	5130 B	3818 B	5829 B	17896 B
SPHINCS/Saber	269 MC	704 B	17760 B	18464 B	18560 B	19264 B	74048 B

Questions?

Joël Alwen, Sandro Coretti, and Yevgeniy Dodis.
The double ratchet: Security notions, proofs, and modularization for the Signal protocol.
In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 129–158. Springer, Heidelberg, May 2019.

Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe.
Post-quantum key exchange - A new hope.
In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 327–343. USENIX Association, August 2016.

Jacqueline Brendel, Marc Fischlin, Felix Günther, Christian Janson, and Douglas Stebila.
Towards post-quantum security for signal's x3dh handshake.
In *SAC 2020*, 2020.
https://eprint.iacr.org/2019/1356.

Jacqueline Brendel, Rune Fiedler, Felix Günther, Christian Janson, and Douglas Stebila.
Post-quantum asynchronous deniable key exchange and the signal handshake.
Cryptology ePrint Archive, Report 2021/769, 2021.

Ward Beullens, Shuichi Katsumata, and Federico Pintore.
Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices.
In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 464–492. Springer, Heidelberg, December 2020.

Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren.

CSI-FiSh: Efficient isogeny based signatures through class group computations.
In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Heidelberg, December 2019.


Daniel J. Bernstein, Tanja Lange, Chloe Martindale, and Lorenz Panny.
Quantum circuits for the CSIDH: Optimizing quantum evaluation of isogenies.
In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 409–441. Springer, Heidelberg, May 2019.


Xavier Bonnetain and André Schrottenloher.
Quantum security analysis of CSIDH.
In Canteaut and Ishai [CI20], pages 493–522.


Jorge Chávez-Saab, Jesús-Javier Chi-Domínguez, Samuel Jaques, and Francisco Rodríguez-Henríquez.
The sqale of csidh: Square-root vélu quantum-resistant isogeny action with low exponents.
Cryptology ePrint Archive, Report 2020/1520, 2020.
https://ia.cr/2020/1520.


Anne Canteaut and Yuval Ishai, editors.
EUROCRYPT 2020, Part II, volume 12106 of *LNCS*. Springer, Heidelberg, May 2020.


Jintai Ding, Scott R. Fluhrer, and Saraswathy RV.
Complete attack on RLWE key exchange with reused keys, without signal leakage.
In Willy Susilo and Guomin Yang, editors, *ACISP 18*, volume 10946 of *LNCS*, pages 467–486. Springer, Heidelberg, July 2018.

Samuel Dobson and Steven D. Galbraith.
Post-quantum signal key agreement with sidh.
Cryptology ePrint Archive, Report 2021/1187, 2021.
https://ia.cr/2021/1187.

Bor de Kock, Kristian Gjøsteen, and Mattia Veroni.
Practical isogeny-based key-exchange with optimal tightness.
In *SAC 2020*, 2020.
https://eprint.iacr.org/2020/1165.

Jintai Ding, Xiang Xie, and Xiaodong Lin.
A simple provably secure key exchange scheme based on the learning with errors problem.
Cryptology ePrint Archive, Report 2012/688, 2012.
https://eprint.iacr.org/2012/688.

Luca De Feo, Samuel Dobson, Steven D. Galbraith, and Lukas Zobernig.
Sidh proof of knowledge.
Cryptology ePrint Archive, Report 2021/1023, 2021.
https://ia.cr/2021/1023.

Scott Fluhrer.
Cryptanalysis of ring-LWE based key exchange with key share reuse.
Cryptology ePrint Archive, Report 2016/085, 2016.
https://eprint.iacr.org/2016/085.

Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti.
On the security of supersingular isogeny cryptosystems.
In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 63–91. Springer, Heidelberg, December 2016.

Keitaro Hashimoto, Shuichi Katsumata, Kris Kwiatkowski, and Thomas Prest.
An efficient and generic construction for signal's handshake (X3DH): Post-quantum, state leakage secure, and deniable.
In Juan Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 410–440. Springer, Heidelberg, May 2021.

Tomoki Kawashima, Katsuyuki Takashima, Yusuke Aikawa, and Tsuyoshi Takagi.
An efficient authenticated key exchange from random self-reducibility on CSIDH.
In Deukjo Hong, editor, *ICISC 20*, volume 12593 of *LNCS*, pages 58–84. Springer, Heidelberg, December 2020.

Xingye Lu, Man Ho Au, and Zhenfei Zhang.
Raptor: A practical lattice-based (linkable) ring signature.
In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19*, volume 11464 of *LNCS*, pages 110–130. Springer, Heidelberg, June 2019.

Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai.
CRYSTALS-DILITHIUM.
Technical report, National Institute of Standards and Technology, 2020.

available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.


Moxie Marlinspike and Trevor Perrin.
The x3dh key agreement protocol, November 2016.
https://signal.org/docs/specifications/x3dh/.


Chris Peikert.
Lattice cryptography for the internet.
In Michele Mosca, editor, *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014*, pages 197–219. Springer, Heidelberg, October 2014.


Chris Peikert.
He gives C-sieves on the CSIDH.
In Canteaut and Ishai [CI20], pages 463–492.


Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang.
FALCON.
Technical report, National Institute of Standards and Technology, 2017.
available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.


Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang.

FALCON.
Technical report, National Institute of Standards and Technology, 2020.
available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.


Ronald L. Rivest, Adi Shamir, and Yael Tauman.
How to leak a secret.
In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, Heidelberg, December 2001.


Nihal Vatandas, Rosario Gennaro, Bertrand Ithurburn, and Hugo Krawczyk.
On the cryptographic deniability of the Signal protocol.
In Mauro Conti, Jianying Zhou, Emiliano Casalicchio, and Angelo Spognardi, editors, *ACNS 20, Part II*, volume 12147 of *LNCS*, pages 188–209. Springer, Heidelberg, October 2020.


Tsz Hon Yuen, Muhammed F. Esgin, Joseph K. Liu, Man Ho Au, and Zhimin Ding.
DualRing: Generic construction of ring signatures with efficient instantiations.
In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 251–281, Virtual Event, August 2021. Springer, Heidelberg.