

IMPROVED STRONGLY DENIABLE AUTHENTICATED KEY EXCHANGES FOR SECURE MESSAGING

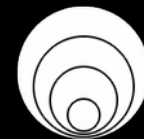
Nik Unger

and

Ian Goldberg

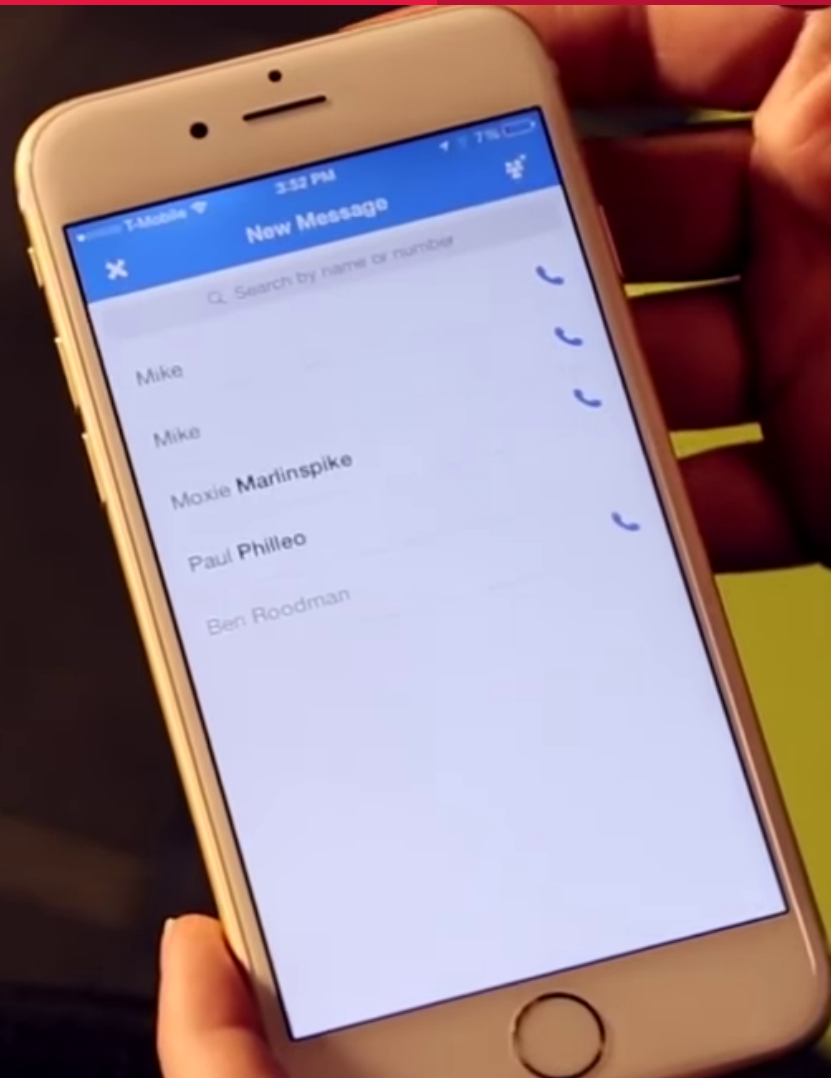


UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS
David R. Cheriton School
of Computer Science

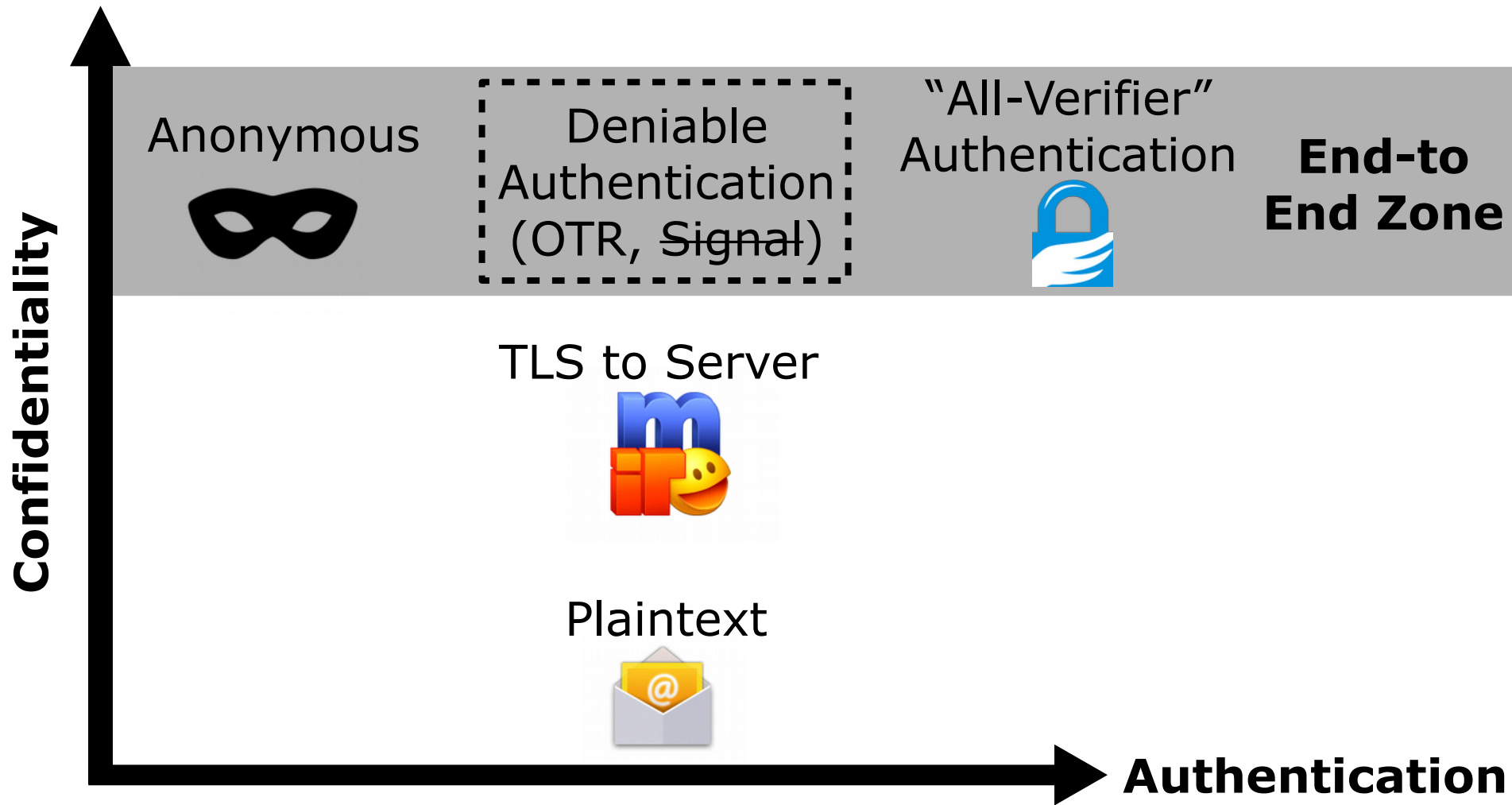


CrySP

Secure Messaging



Secure Messaging



Why Deniability?

HERE'S CRYPTOGRAPHIC PROOF THAT DONNA BRAZILE IS WRONG, WIKILEAKS EMAILS ARE REAL

Luke Rosiak | Investigative Reporter

7:10 PM 10/21/2016

Cryptographic signatures demonstrate that Democratic National Committee Chairman Donna Brazile is wrong when she suggests the WikiLeaks emails were altered and that she did not send an email tipping off Democratic presidential nominee Hillary Clinton to debate questions.

Many email systems use a verification system called DomainKeys Identified Mail (DKIM) that shows whether an email has been changed. It uses a key stored on the email server that sent the email, so it can't be forged.

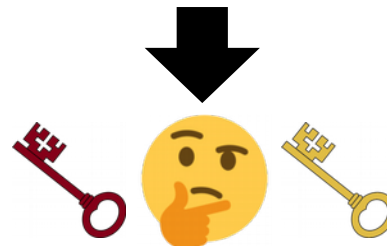
HillaryClinton.com uses Gmail to handle its mail and uses DKIM. Staffer Jennifer Palmieri, using her HillaryClinton.com email, replied to a Brazile email warning that Brazile was "worried" about Clinton's ability to answer a question about the

Deniable Messaging



```
ad a4 5a 73 71 38 98 09 e4 db 58 40 04 97 a2 44 ..Zsq8...x@...D
39 03 da 96 e9 2e 61 4f 17 87 41 c2 7e 1a 27 9b 9.....a0...A.~...'
2d 7f 1e 10 19 eb 1b 53 17 70 8c b9 e2 14 84 85 -.5 p.....
d2 1e 9b ca 59 f7 cb 8f ec db 94 e1 3a 8f 2a b8 .....Y...:.*
b8 89 8c 42 98 80 70 86 05 21 fb 2d b4 de 8a c5 ...B.p. !.-....
bc 8e bb d3 23 f8 9c ad da bf 3c 86 e4 cb 2b df .....#.....<...+
fb b5 1c a4 12 78 42 b3 db 36 c5 6e a2 af bd 3c .....xB. .6.n...<
dc 77 c1 10 5c a8 c0 47 d6 d7 f9 6e 6e 00 a2 59 .w.\..G ..nn..Y
b8 6b 32 95 76 5f e2 29 7e 68 18 7a 93 b8 62 ee .k2.v..) ~h.z..b.
7d 1f 2c 72 ba 81 98 67 0f 9d a7 50 fb 54 77 02 }.r...g ...P.Tw.
9e 74 87 d5 4f 8c e1 91 83 d3 4d 1c 3c c4 5a bd .t..O...M.<.Z.
ec d9 29 d5 16 7f 2b 52 c9 45 27 dc 8b fe 8c 47 ..)...+R .E'....G
e6 1c 18 23 ef 2e e2 3e e3 4b ba 99 46 45 00 d3 ...#...> .K..FE..
19 03 27 f4 e6 53 08 10 d6 c0 5c 40 99 bc 2e 6f .....S... \@...o
99 bd 13 a3 d7 51 cc 74 18 88 9a 99 f5 f7 d3 e4 .....Q.t .....
6d ac 86 93 1f b0 6f 72 37 99 ca ac 26 ac f9 47 m....or 7...&..G
7d 92 87 23 13 d2 39 2c 98 95 3b 99 05 80 de 00 }.#..9, ;;.....
8d 52 97 94 b3 dc 54 43 c6 9d 5e 46 87 08 78 84 .R....TC ...^F..x.
1c d2 32 09 e6 35 56 06 08 0f ff 34 a3 24 b8 e6 ..2..5V. ...4.$..
aa ce 91 9a 5f 72 d5 17 0b 00 fb 32 6a fc 9e c4 .....r... ..2j...
ba 51 22 55 63 a5 fb 2d 5e 86 6b 0c f9 f2 3e 25 .Q"UC...- ^.k...>%
```

 there's a protest about it tomorrow
 want to go?
<A> Yes!
 ok, no phones



Deniable Messaging

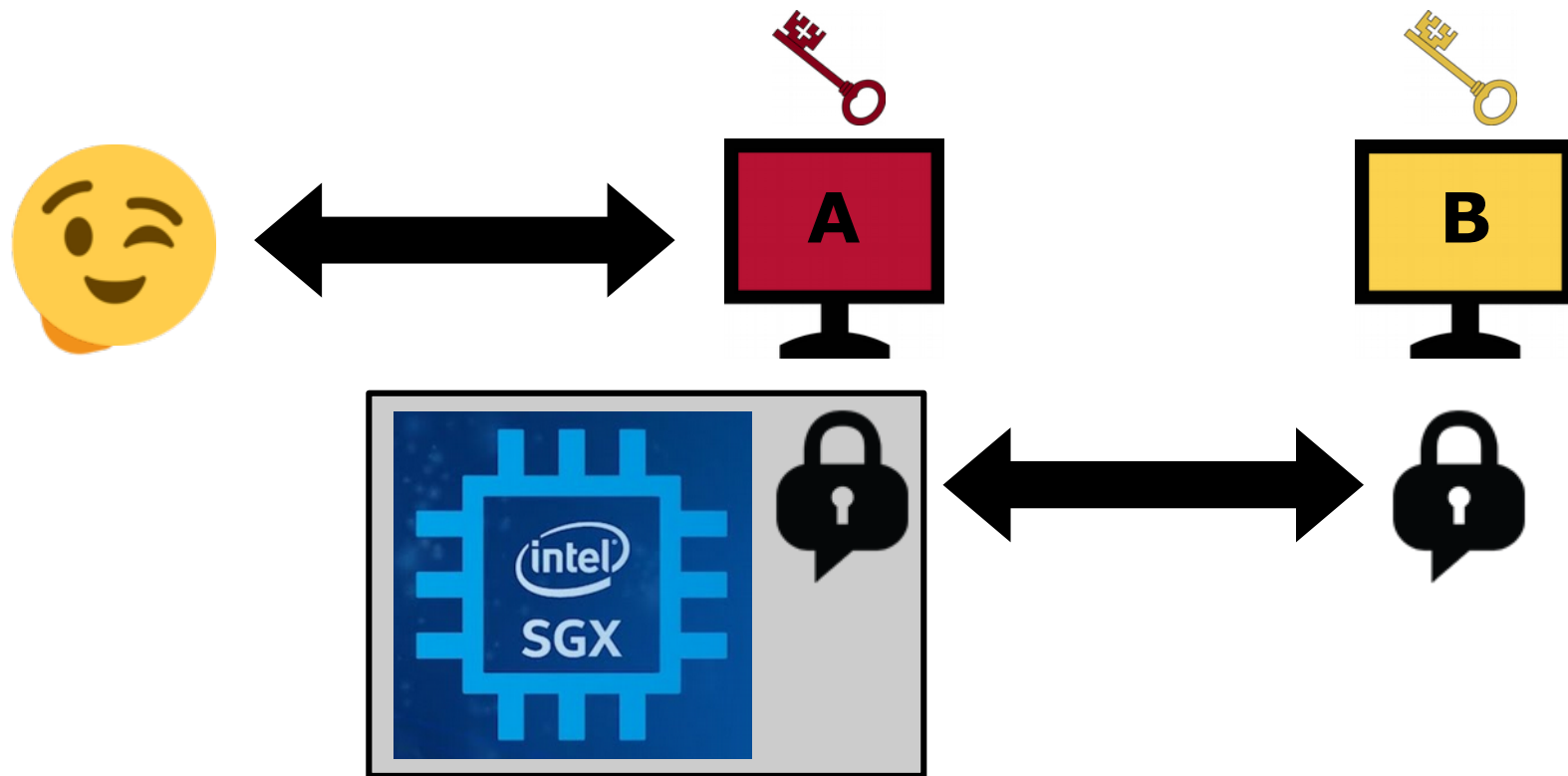


```
ad a4 5a 73 71 38 98 09 e4 db 58 40 04 97 a2 44 ..Zsq8.. ..x@...D
39 03 da 96 e9 2e 61 4f 17 87 41 c2 7e 1a 27 9b 9.....a0 ..A.~..'
2d 7f 1e 10 19 eb 1b 53 17 70 8c b9 e2 14 84 85 -. ....5 .p.....
d2 1e 9b ca 59 f7 cb 8f ec db 94 e1 3a 8f 2a b8 .....Y.....:.*
b8 89 8c 42 98 80 70 86 05 21 fb 2d b4 de 8a c5 ...B..p. !.-....
bc 8e bb d3 23 f8 9c ad da bf 3c 86 e4 cb 2b df .....#.....<...+
fb b5 1c a4 12 78 42 b3 db 36 c5 6e a2 af bd 3c .....xB. .6.n...<
dc 77 c1 10 5c a8 c0 47 d6 d7 f9 6e 6e 00 a2 59 .w..\..G ...nn..Y
b8 6b 32 95 76 5f e2 29 7e 68 18 7a 93 b8 62 ee .k2.v_.) ~h.z..b.
7d 1f 2c 72 ba 81 98 67 0f 9d a7 50 fb 54 77 02 }. ,r...g ...P.Tw.
9e 74 87 d5 4f 8c e1 91 83 d3 4d 1c 3c c4 5a bd .t..O... .M.<.Z.
ec d9 29 d5 16 7f 2b 52 c9 45 27 dc 8b fe 8c 47 ..)...+R .E'....G
e6 1c 18 23 ef 2e e2 3e e3 4b ba 99 46 45 00 d3 ...#...> .K..FE..
19 03 27 f4 e6 53 08 10 d6 c0 5c 40 99 bc 2e 6f .....S... \@...o
99 bd 13 a3 d7 51 cc 74 18 88 9a 99 f5 f7 d3 e4 .....Q.t .....
6d ac 86 93 1f b0 6f 72 37 99 ca ac 26 ac f9 47 m.....or 7...&..G
7d 92 87 23 13 d2 39 2c 98 95 3b 99 05 80 de 00 }.#..9, ;;.....
8d 52 97 94 b3 dc 54 43 c6 9d 5e 46 87 08 78 84 .R....TC ...^F..x.
1c d2 32 09 e6 35 56 06 08 0f ff 34 a3 24 b8 e6 ..2..5V. ...4.$..
aa ce 91 9a 5f 72 d5 17 0b 00 fb 32 6a fc 9e c4 .....r... ..2j...
ba 51 22 55 63 a5 fb 2d 5e 86 6b 0c f9 f2 3e 25 .Q"UC...- ^.k...>%
```

 there's a protest about it tomorrow
 want to go?
<A> Yes!
 ok, no phones

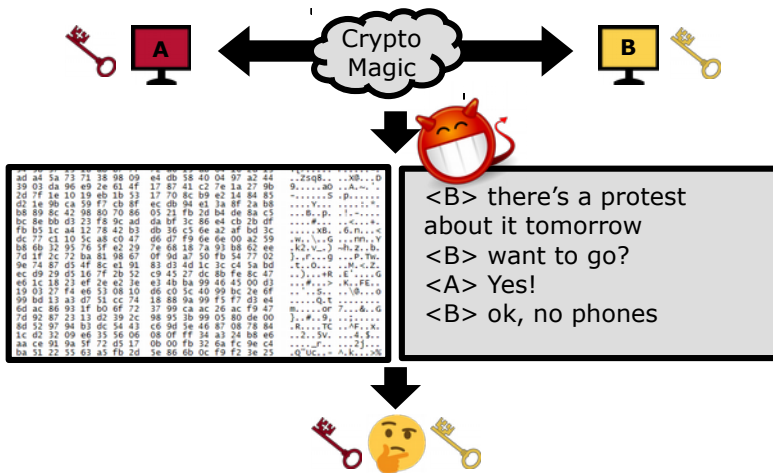


Deniable Messaging...?

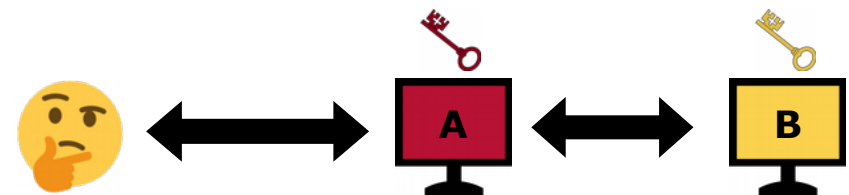


Offline vs. Online Deniability

Offline Deniability



Online Deniability



Deniable Messaging...?

- See Appendix A
 - Attacks on OTRv3 and Signal
- Also see ia.cr/2018/424:

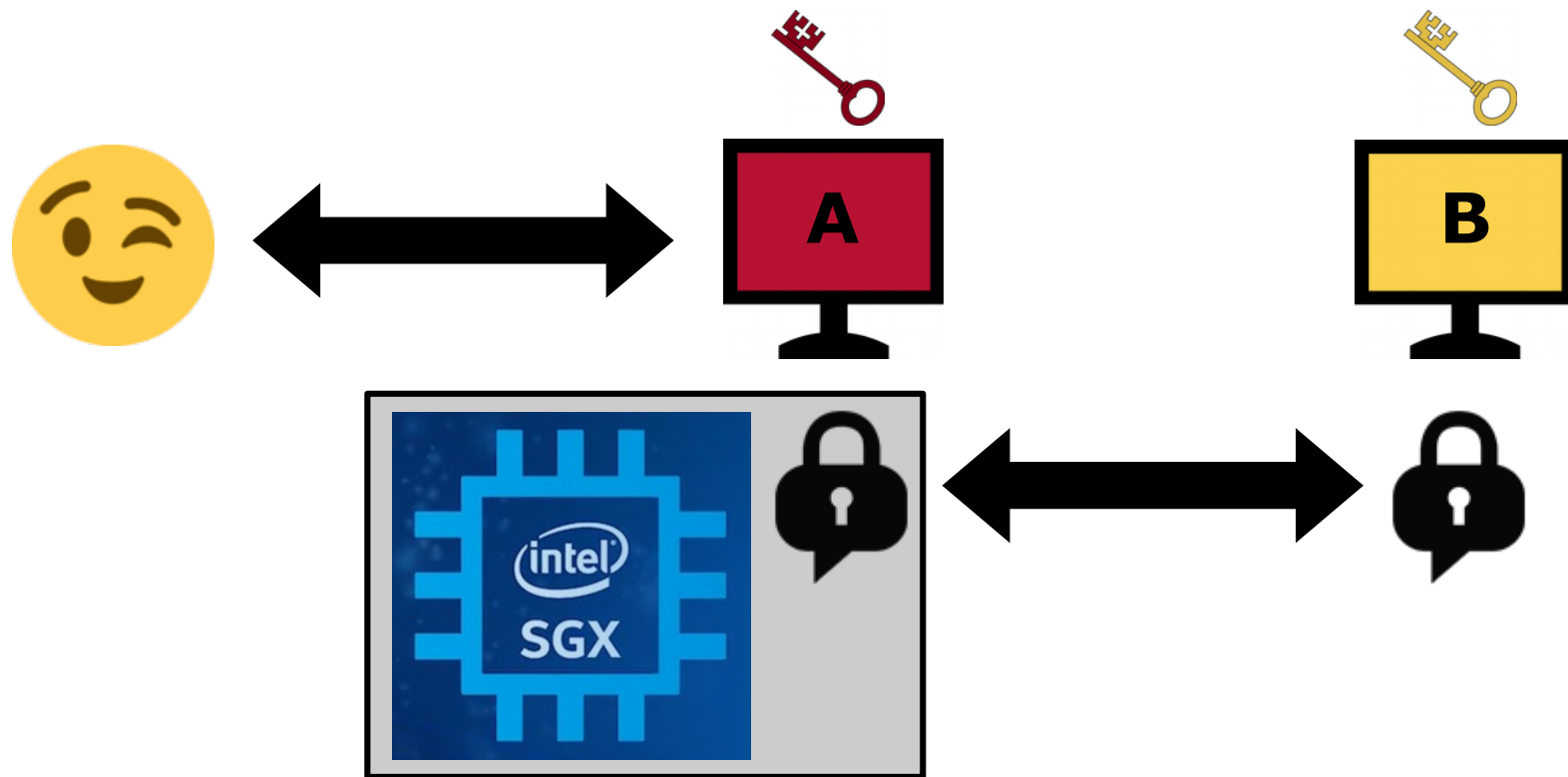
On The Use of Remote Attestation to Break and Repair Deniability

Lachlan J. Gunn
Aalto University
lachlan.gunn@aalto.fi

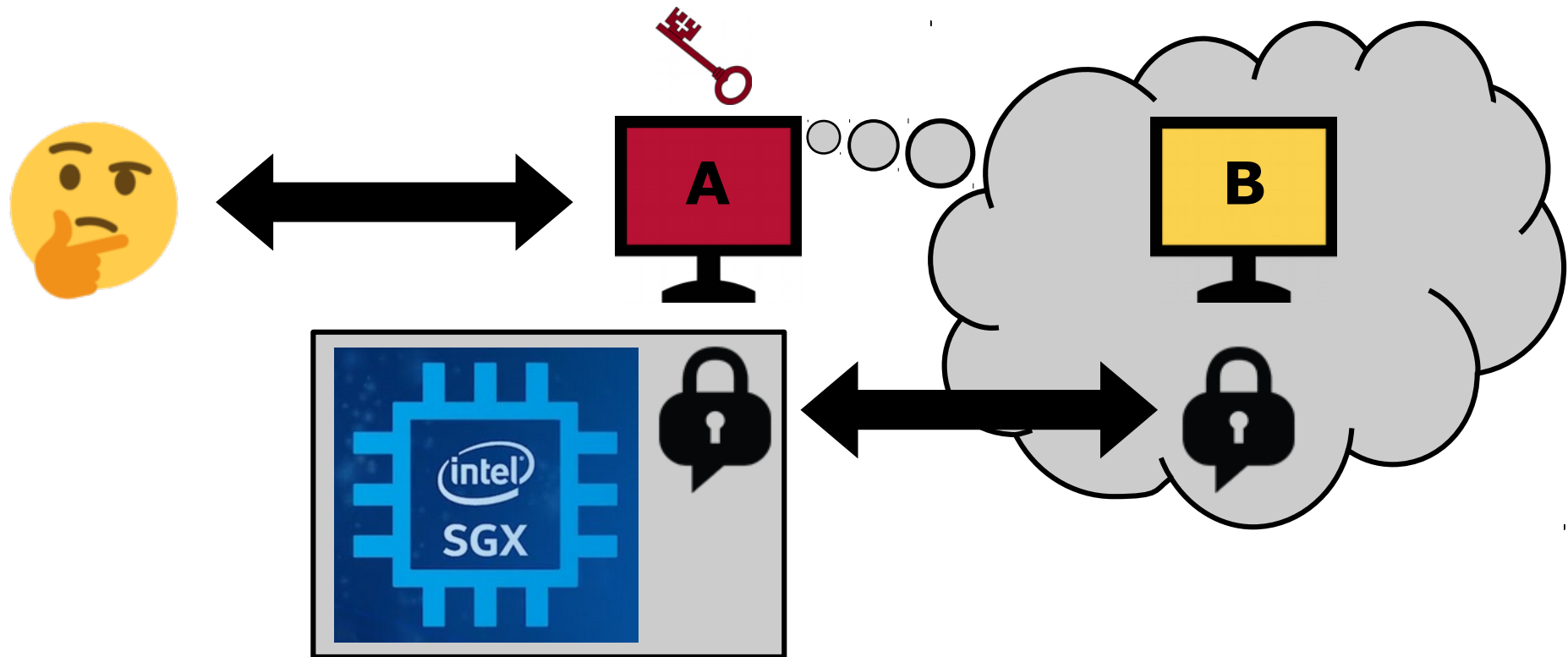
Ricardo Vieitez Parra
Aalto University
ricardo.vieitezparra@aalto.fi

N. Asokan
Aalto University
asokan@acm.org

Deniable Messaging

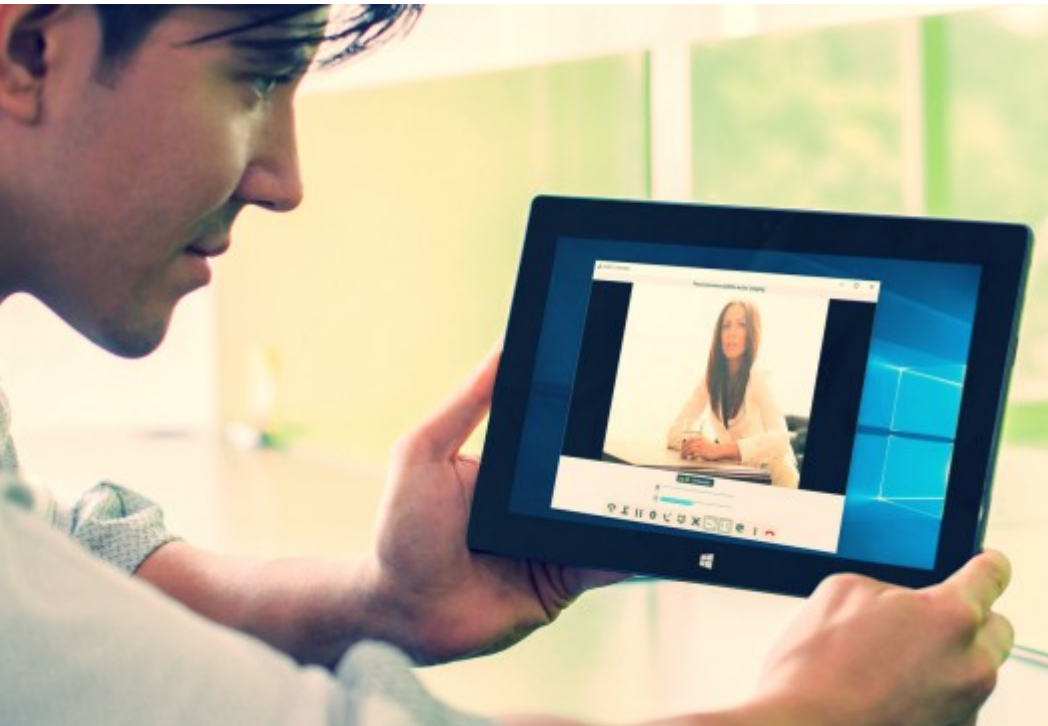


Deniable Messaging

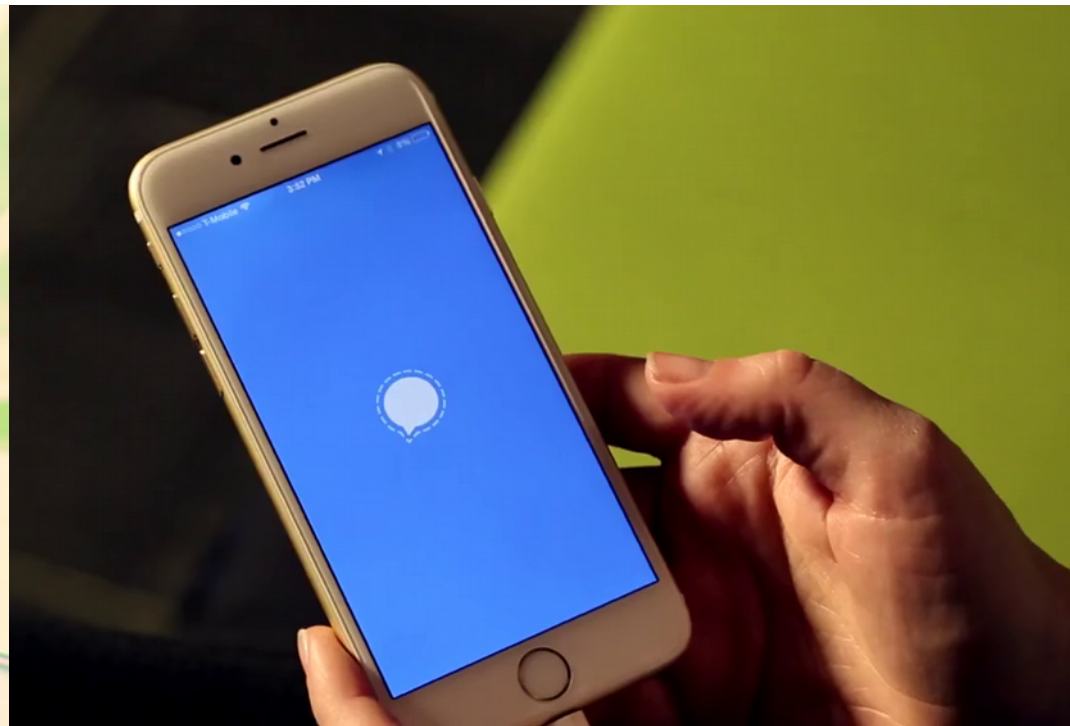


In This Paper

- Two new efficient key exchange protocols



Interactive

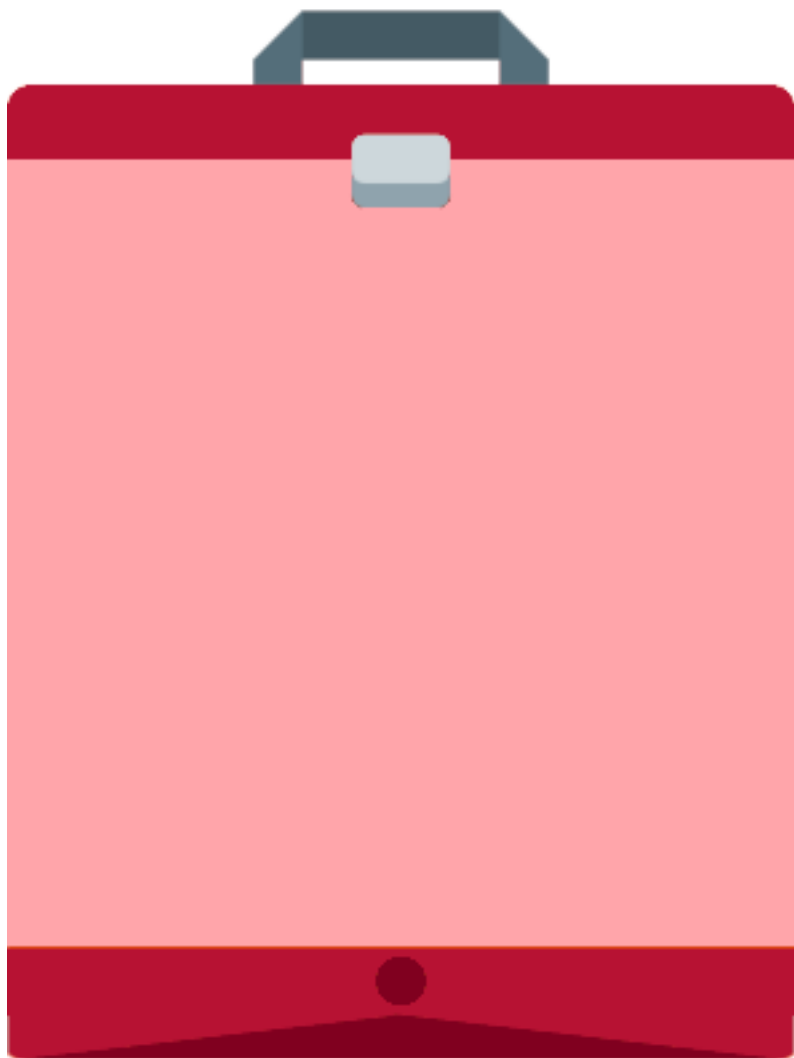


Non-interactive

Security Properties

- Confidentiality
- Mutual authentication
- Forward secrecy
- Contributiveness
- Offline and online deniability

Crypto Toolbox



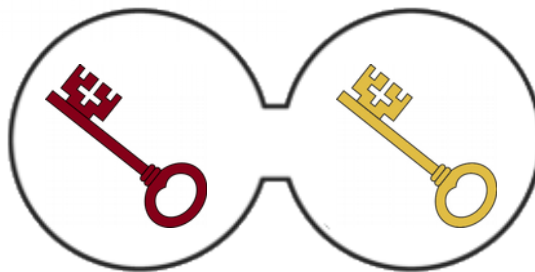
Identity key
(long-term asymmetric)



Ephemeral key
(short-term asymmetric)

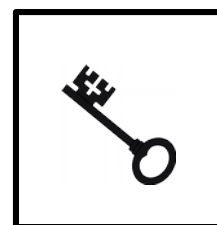
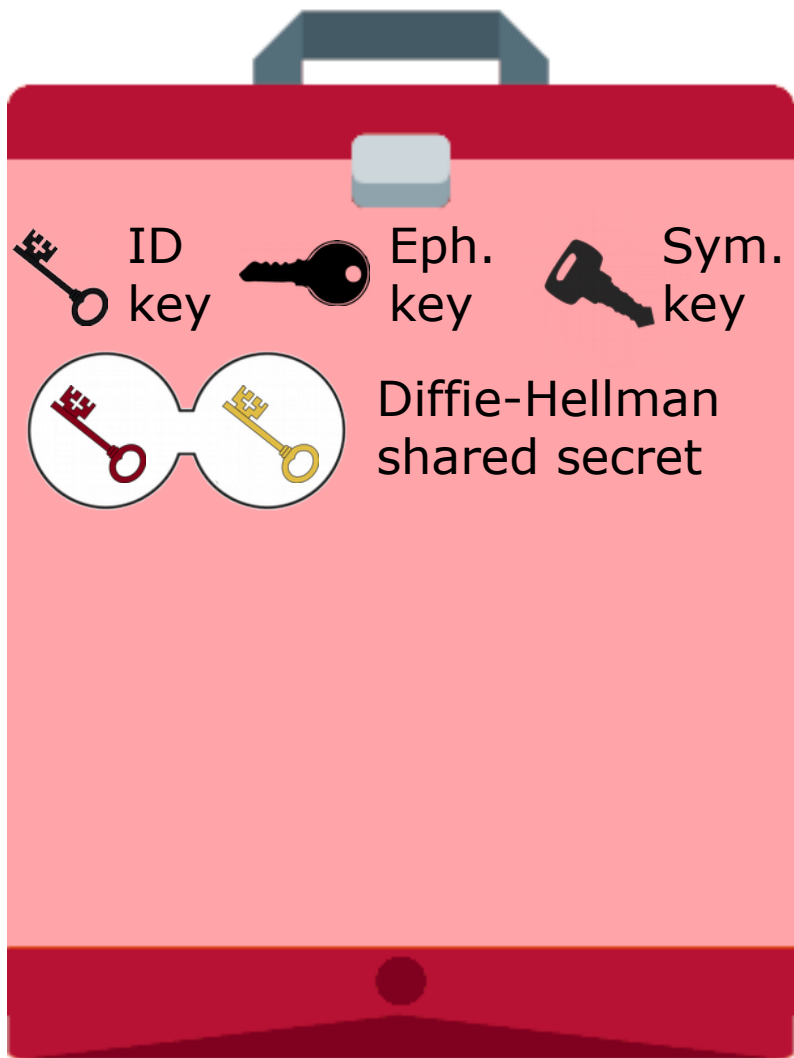


Shared session key
(symmetric)





Diffie-Hellman
shared secret

Crypto Toolbox





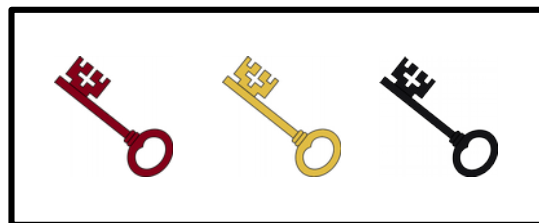
Signature

Create: need private 
Verify: need public 



MAC

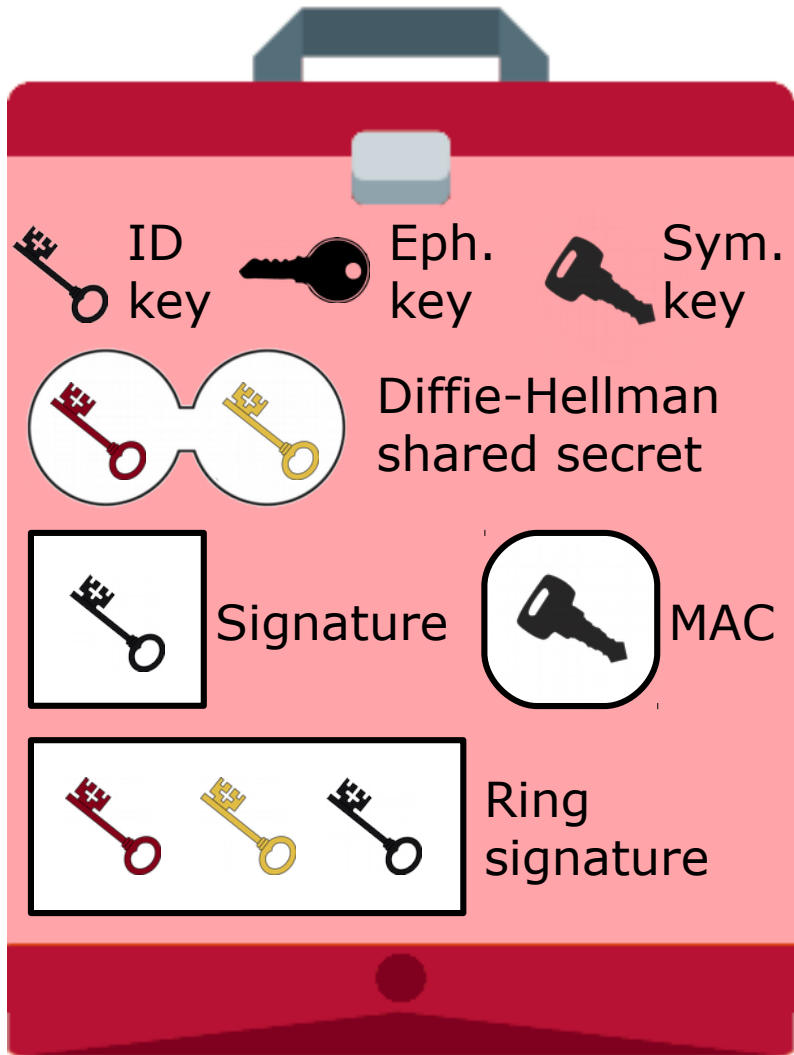
Create: need 
Verify: need 



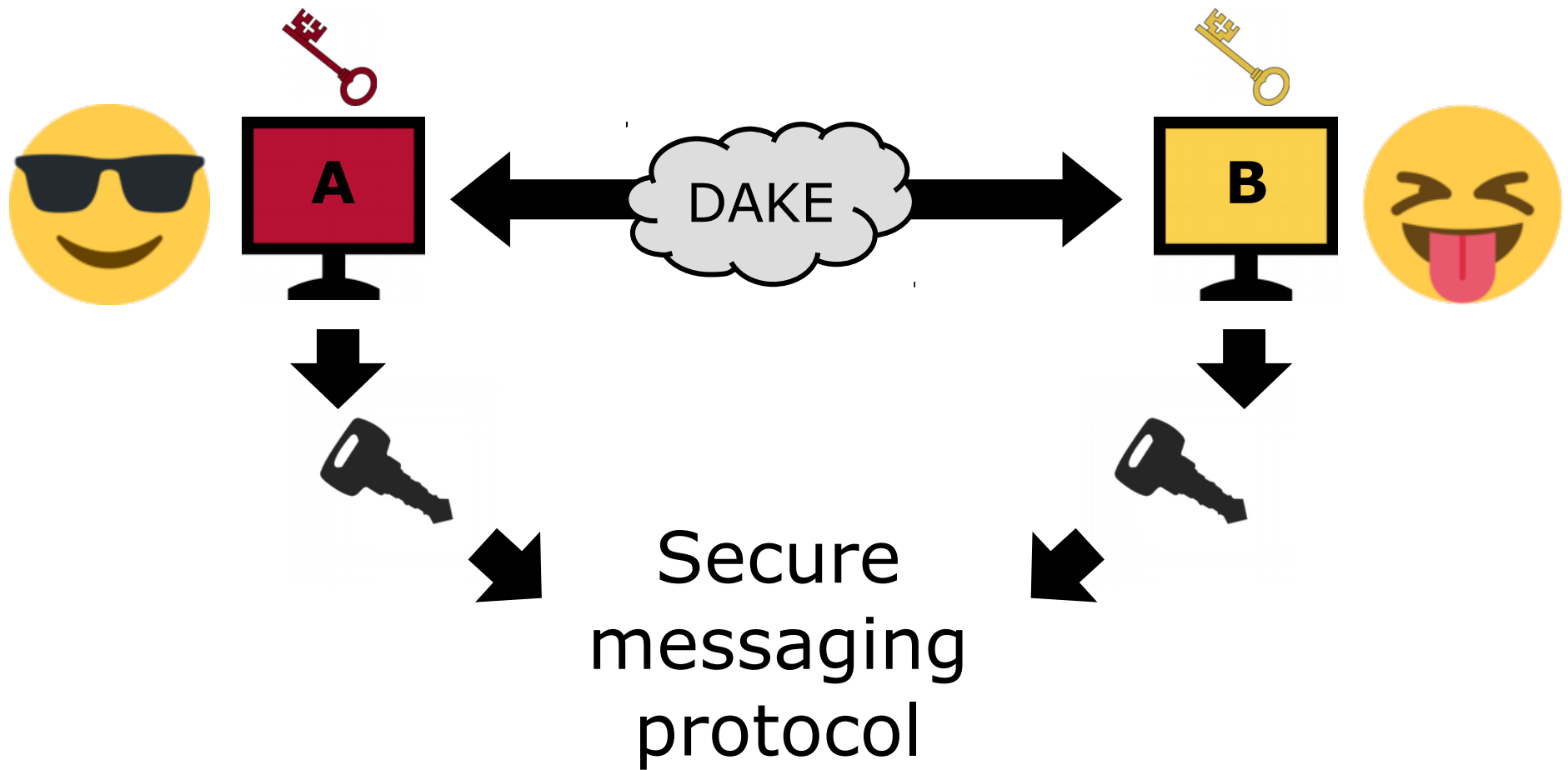
Ring signature

Create: need **one** private 
Verify: need **all** public 
 ,  , or 
 ,  , and 

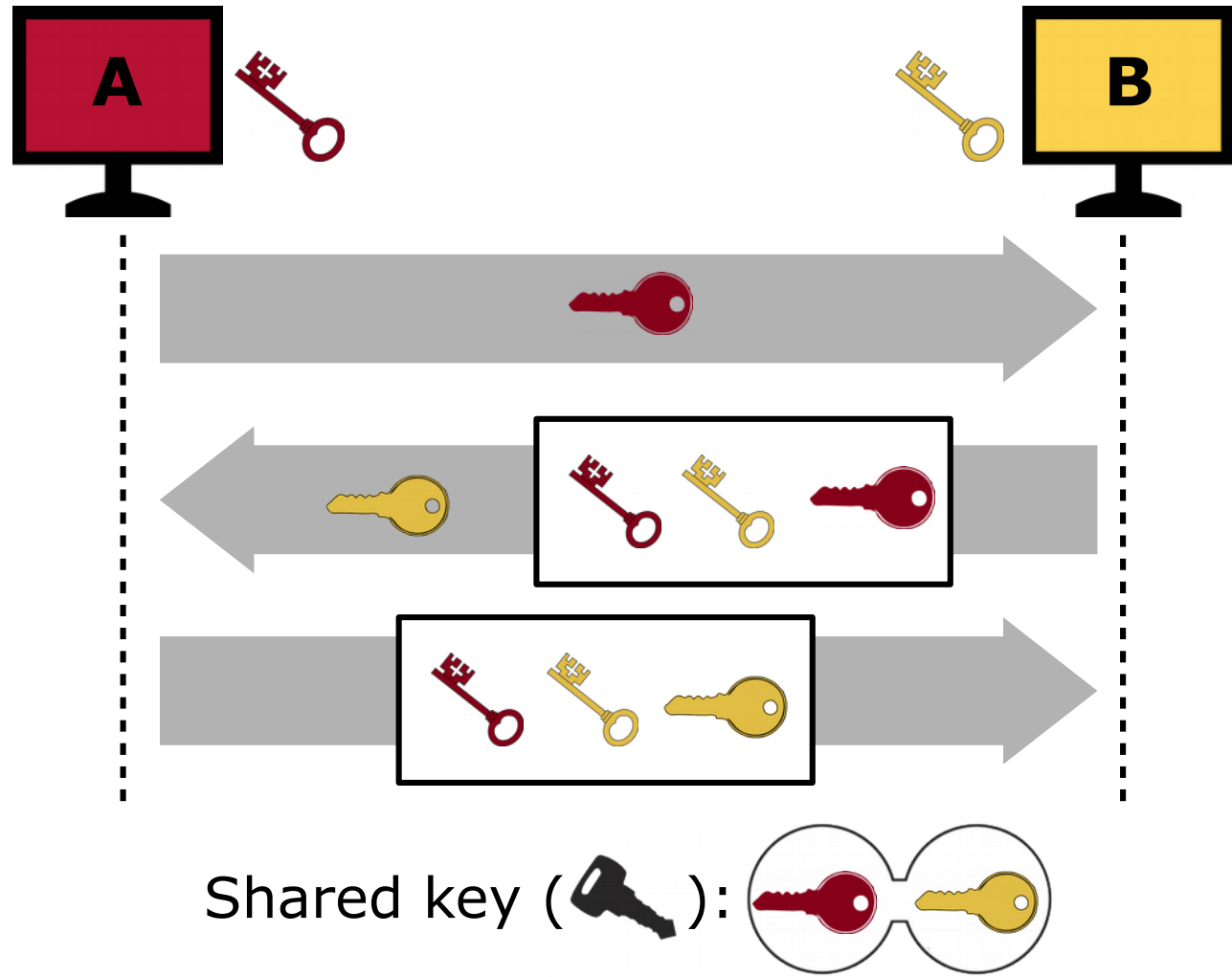
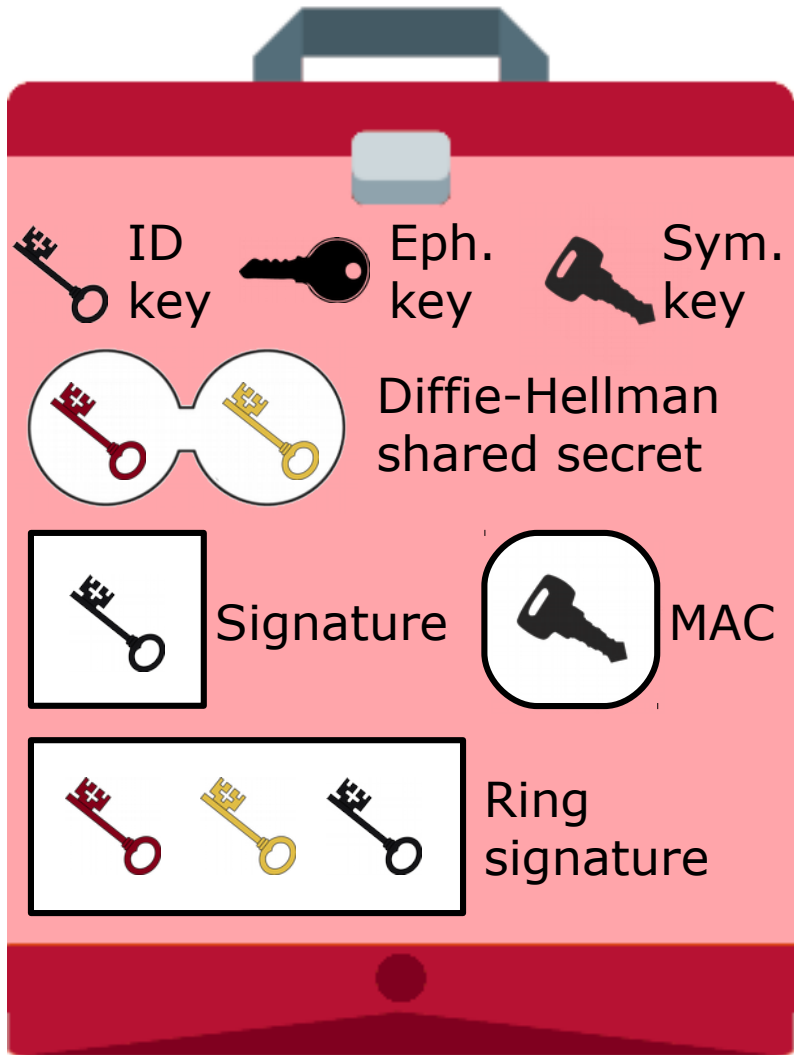
Crypto Toolbox



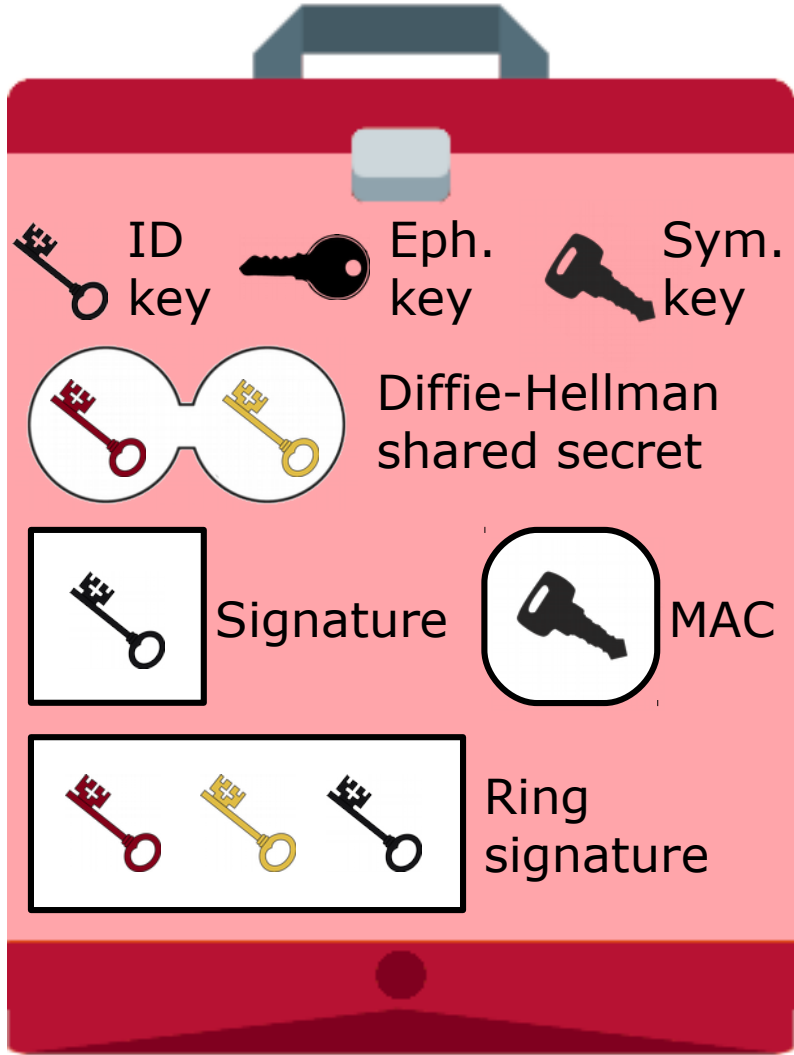
Deniable Authenticated Key Exchanges



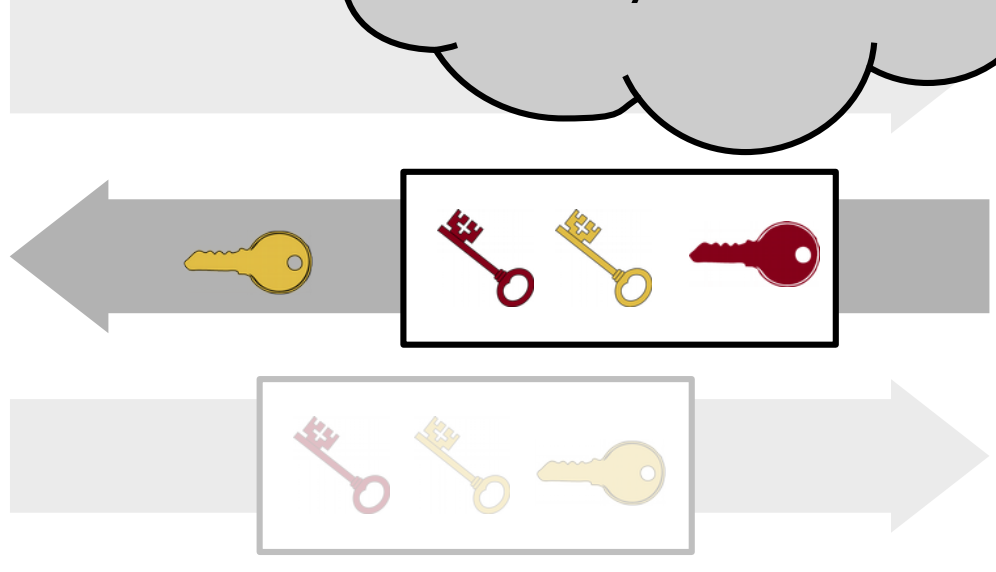
DAKEZ



DAKEZ: Authentication

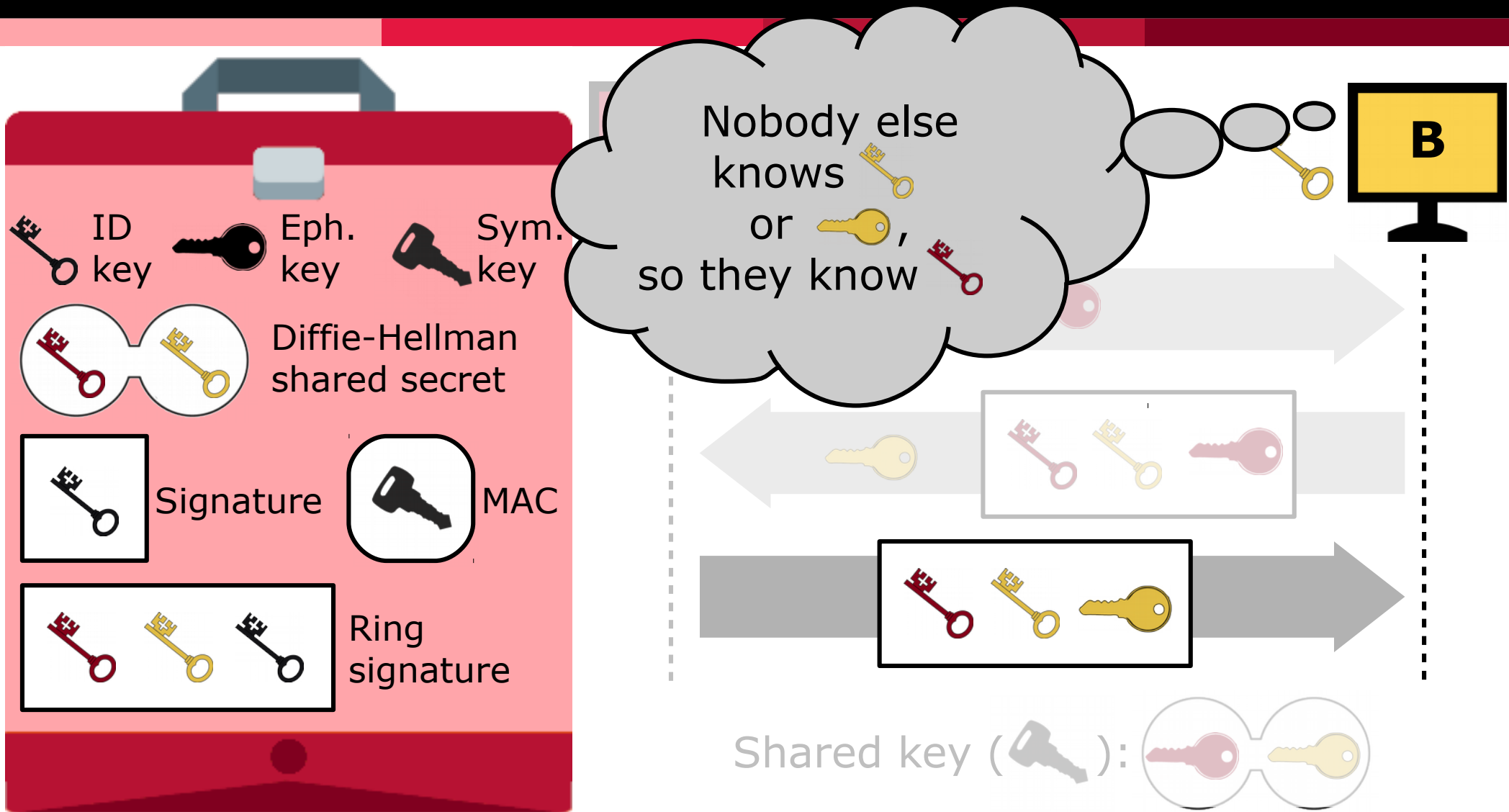


Nobody else knows
or ,
so they know

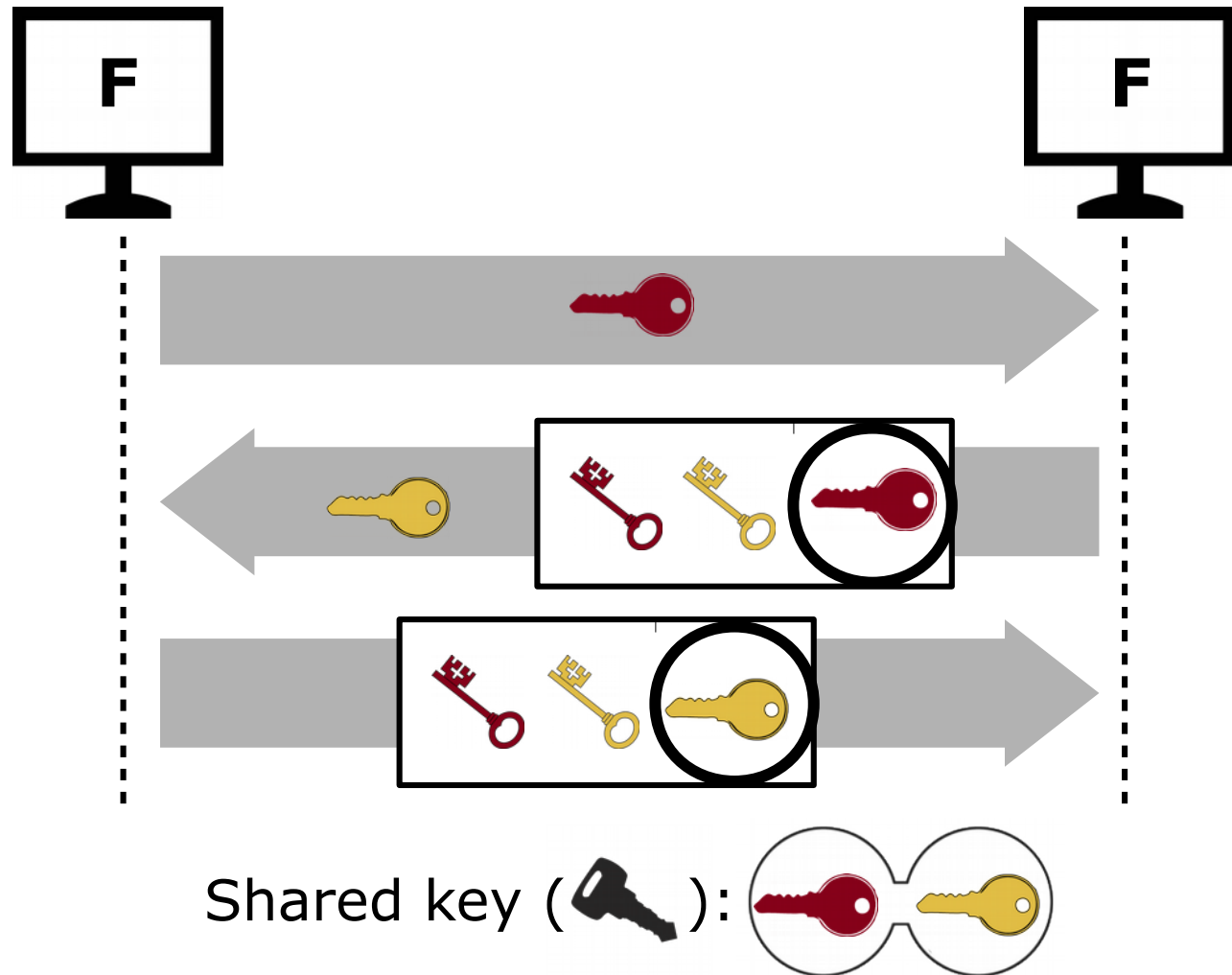
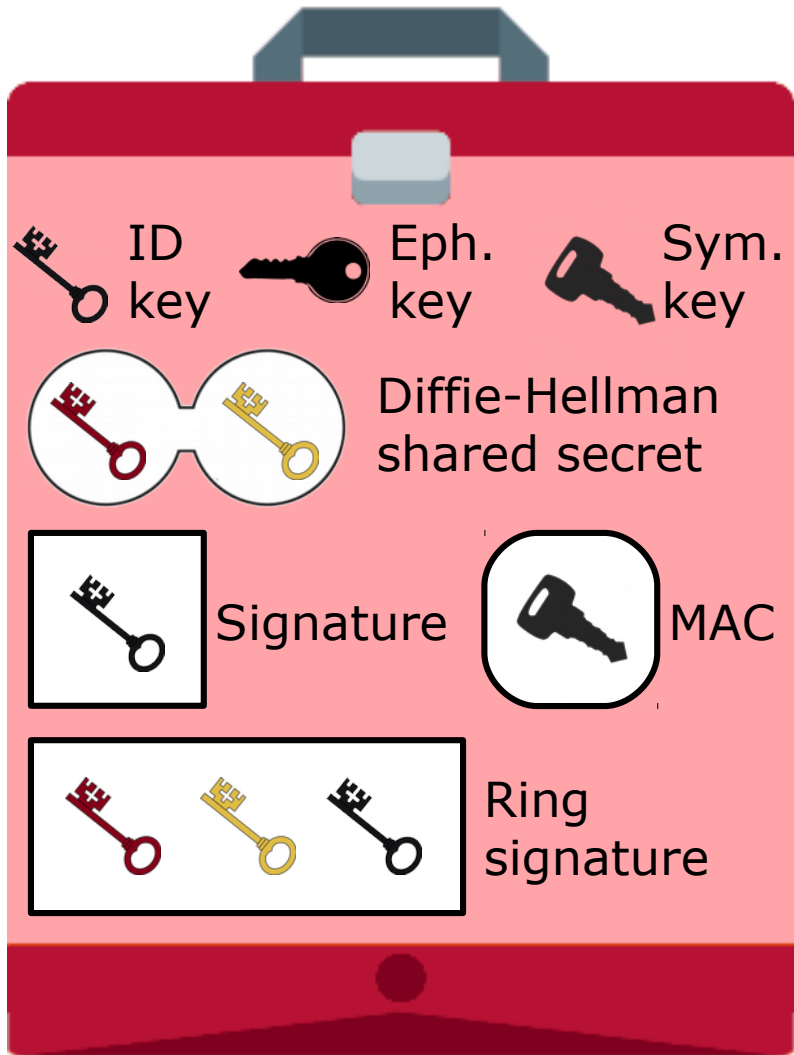


Shared key (): (red key, yellow key)

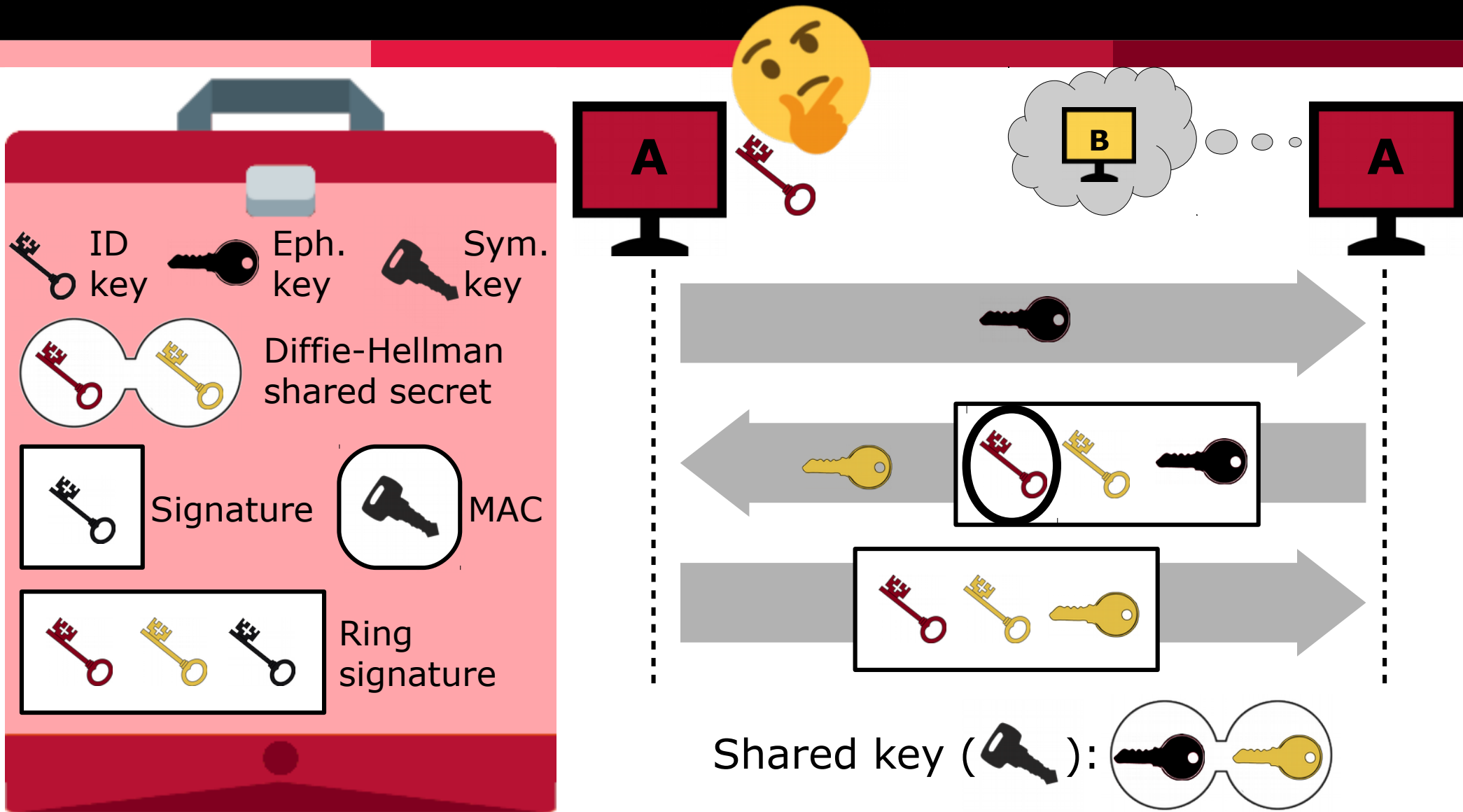
DAKEZ: Authentication



DAKEZ: Offline Deniability



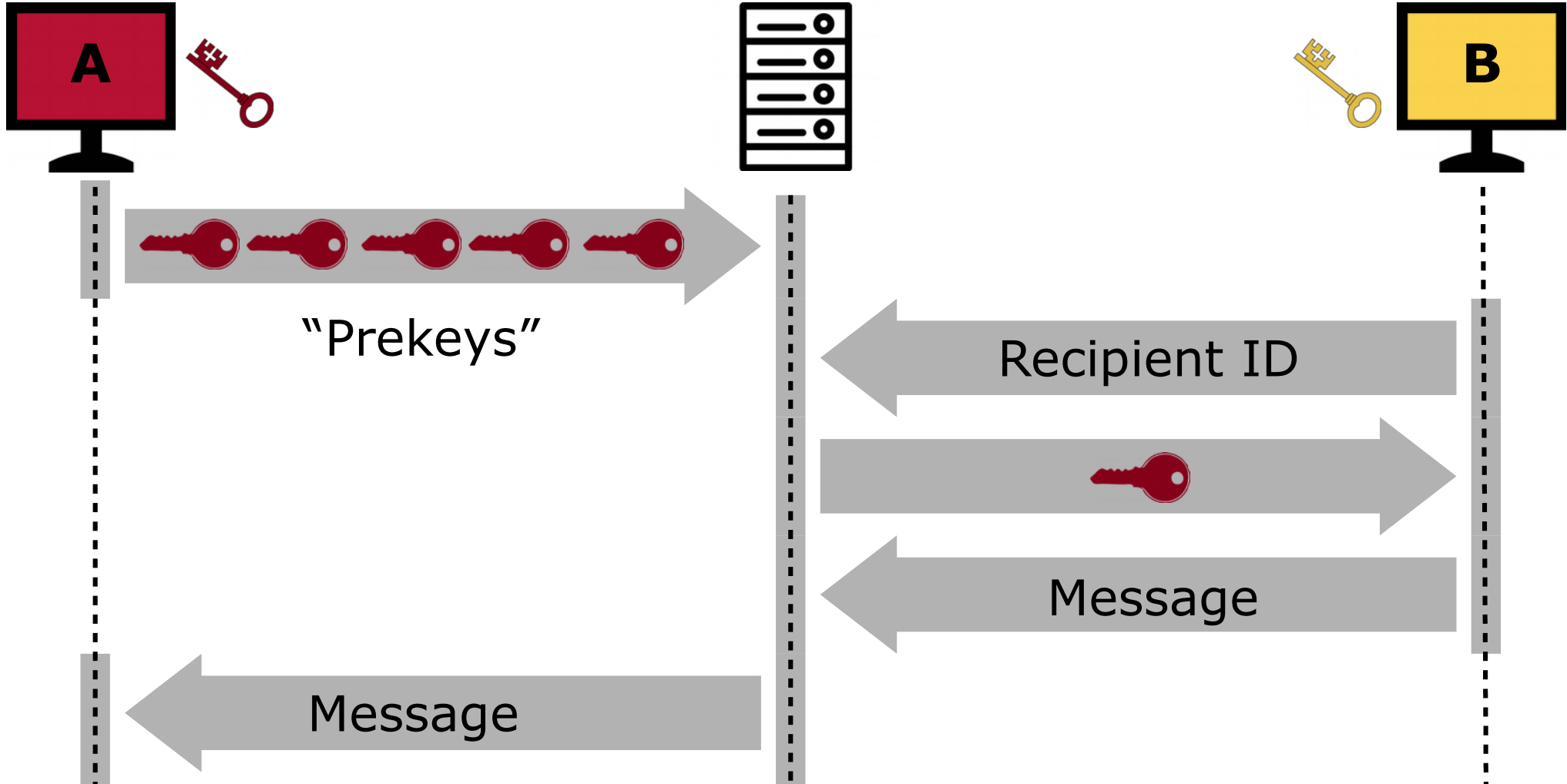
DAKEZ: Online Deniability



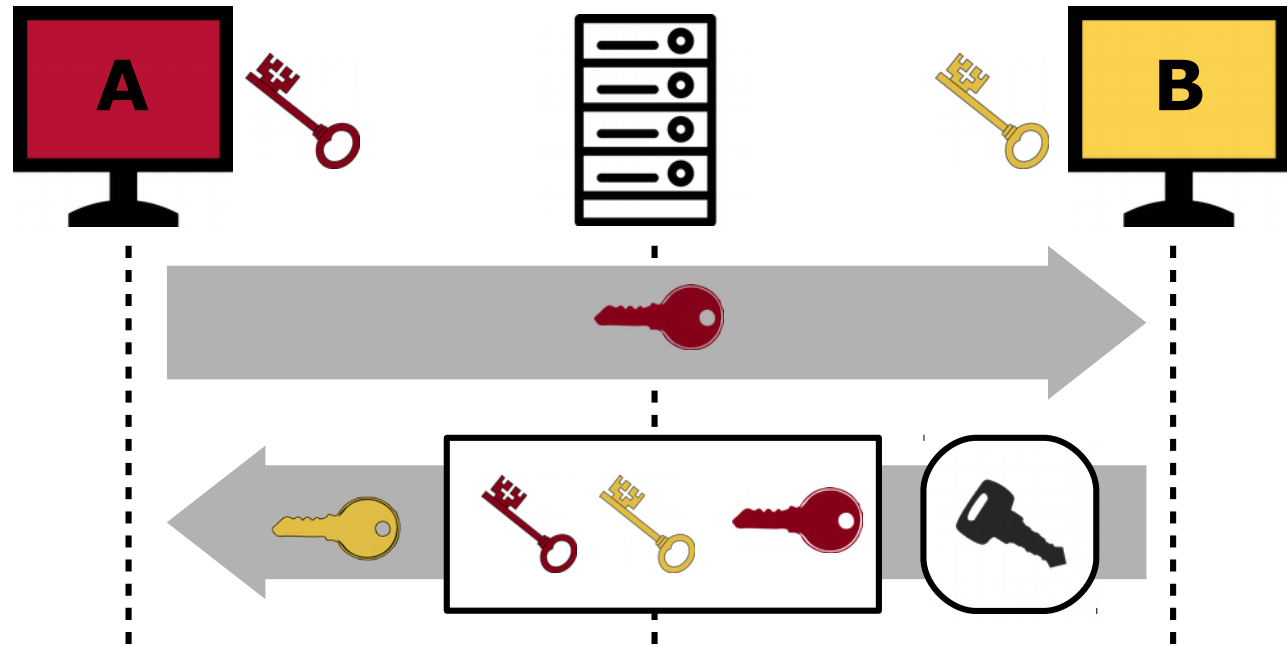
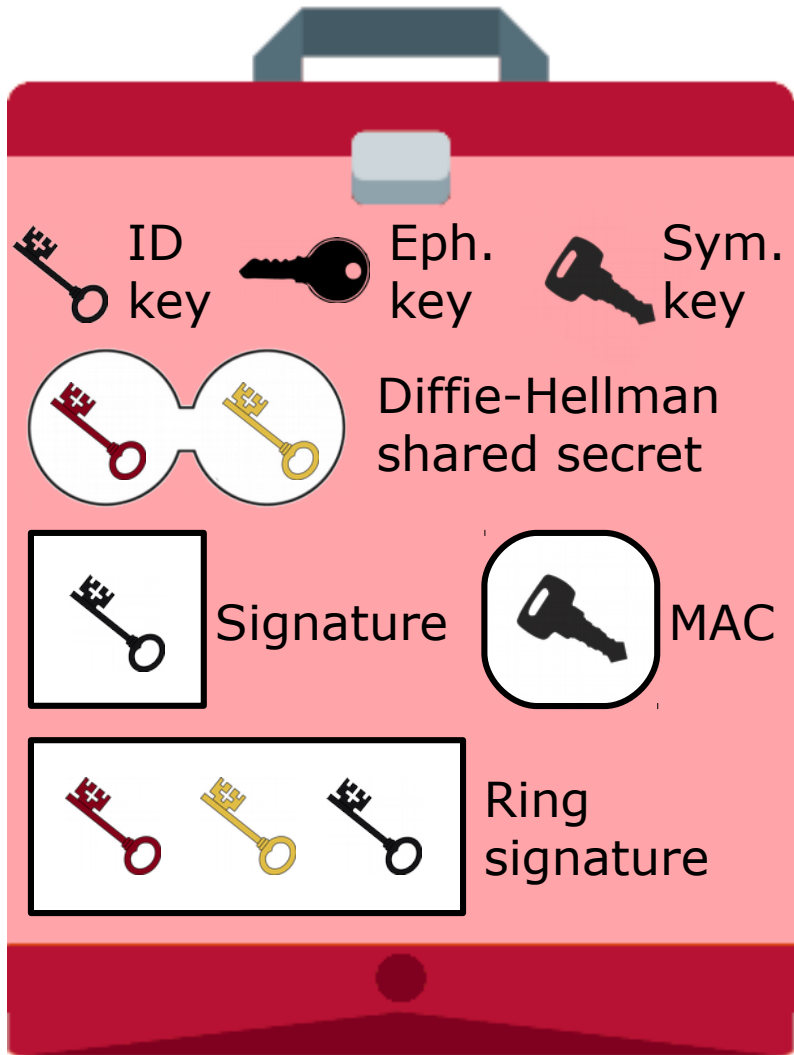
Mobile?



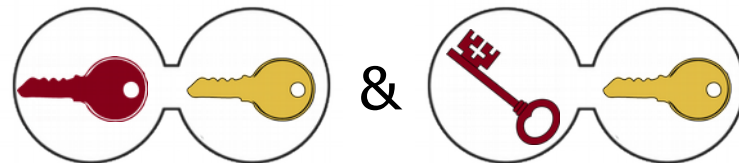
Mobile Use



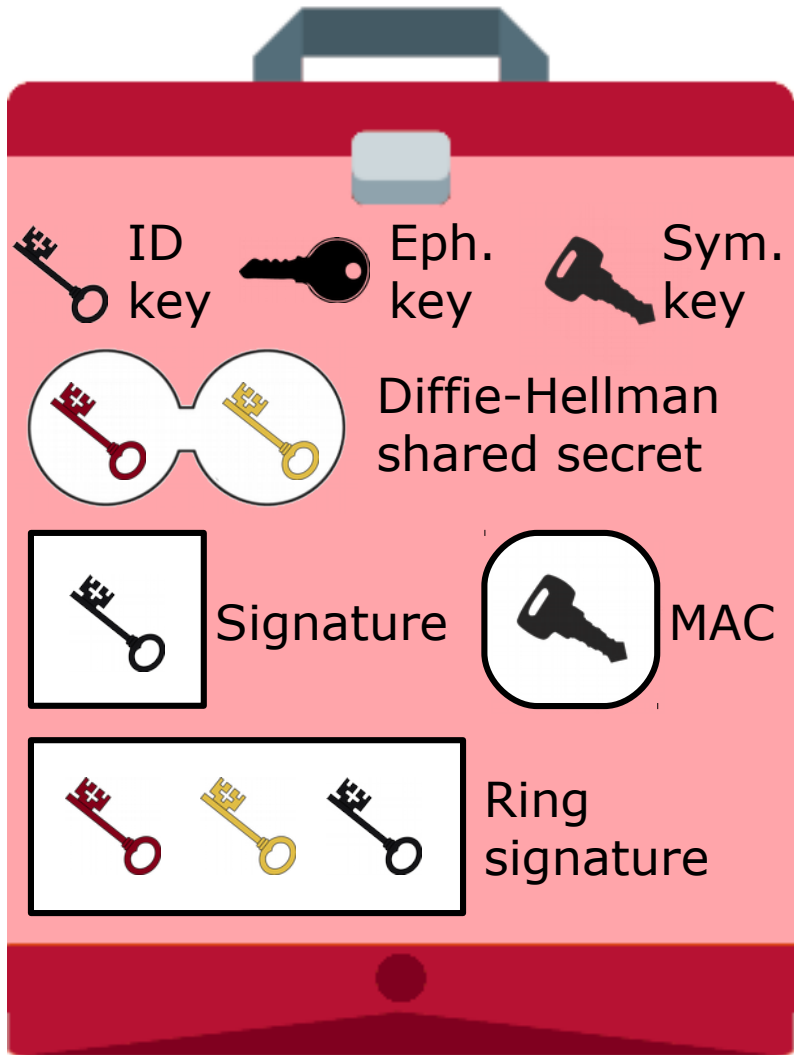
ZDH



Shared key ():



ZDH: Authentication

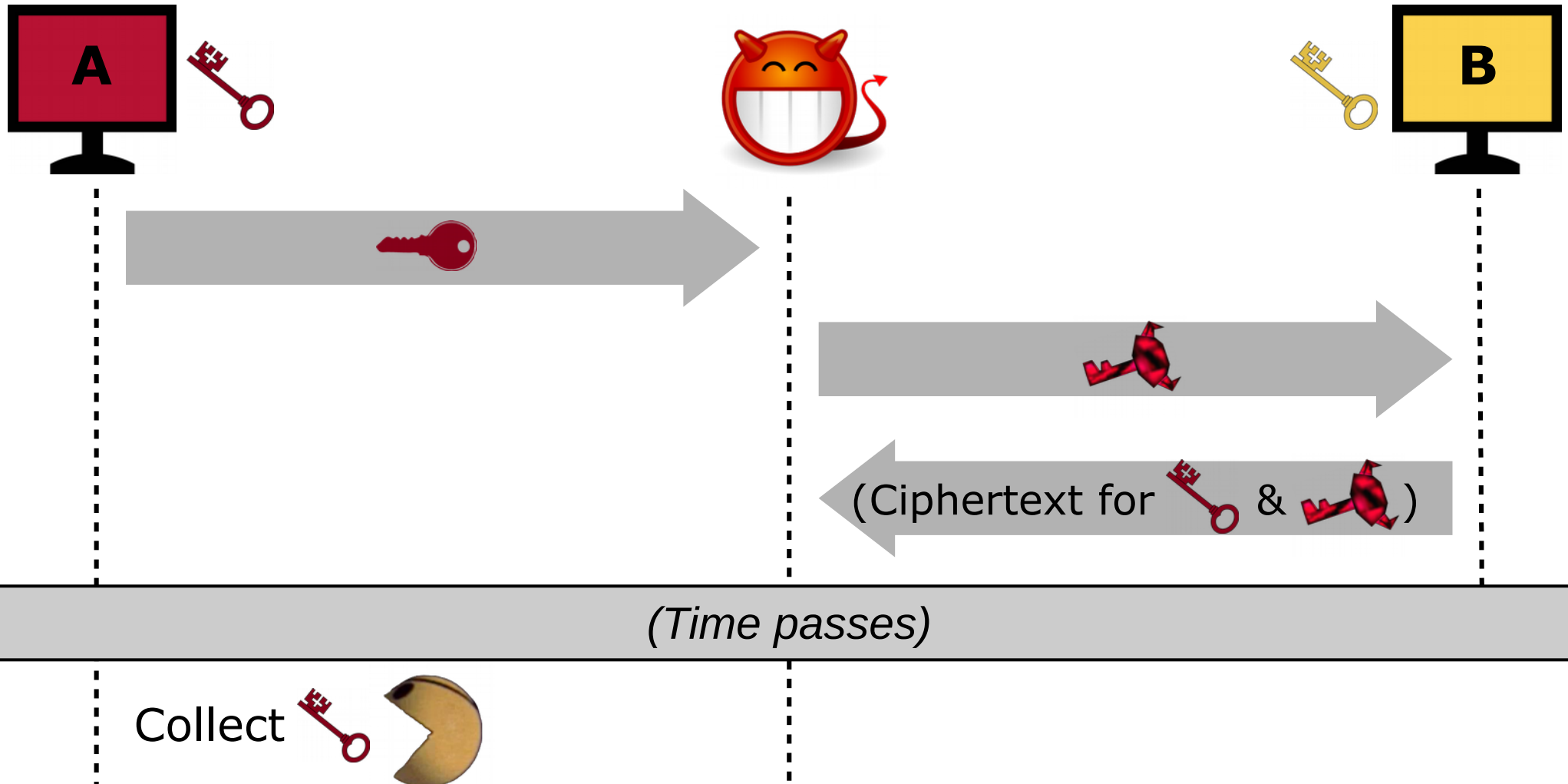


Shared key ():

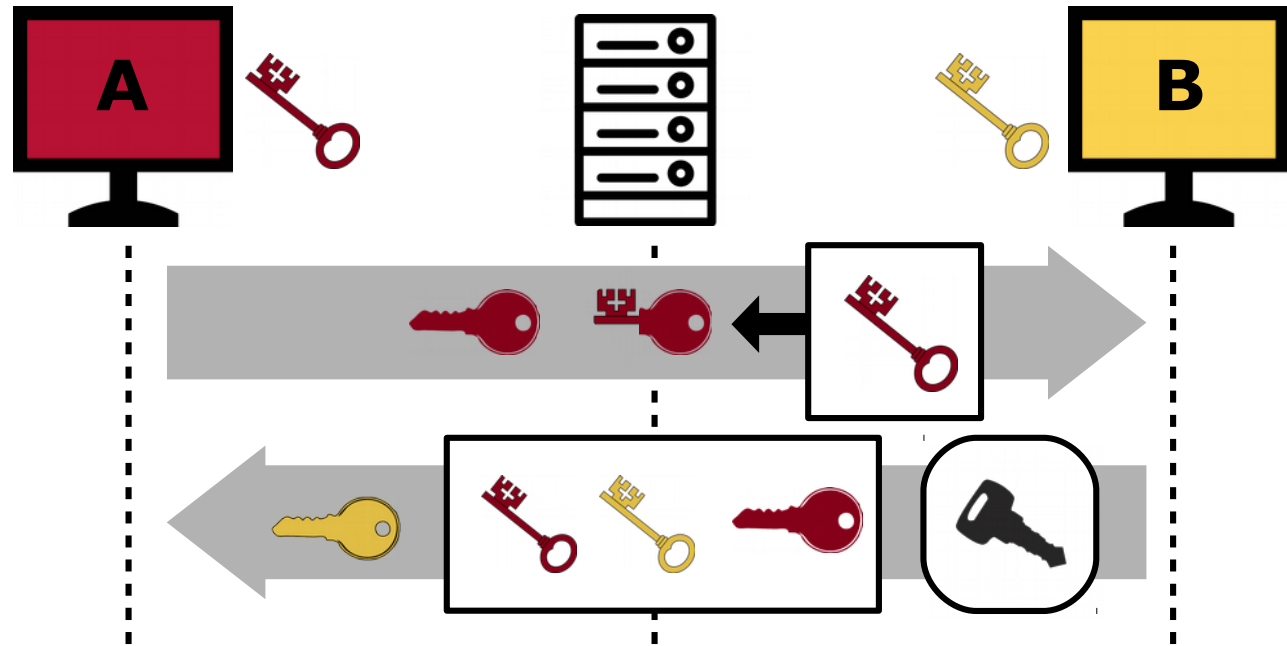
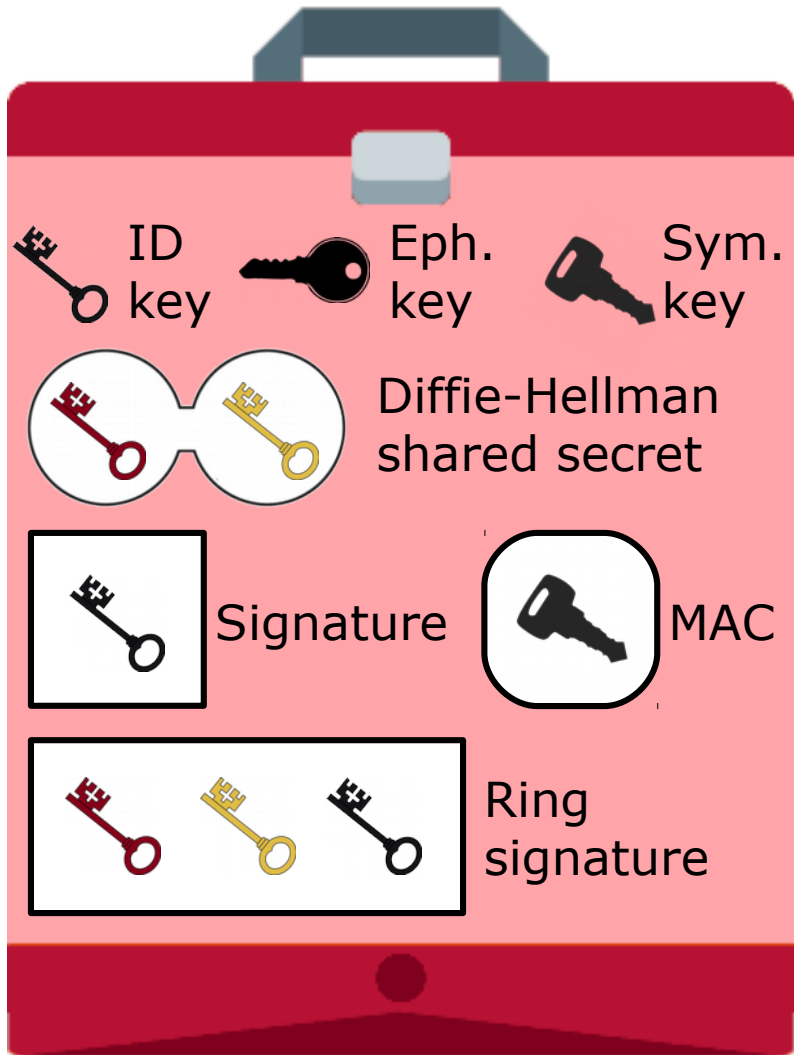


Weak Forward Secrecy

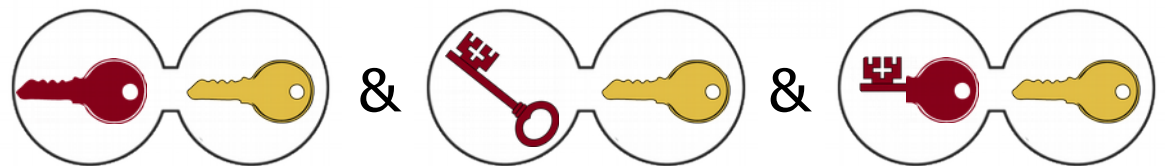
(Like Signal, originally)



XZDH



Shared key ():



Is This Secure?



Is This Secure?

A.0 On the other hand, this attack is not effective.

protections. The attacker can't distinguish between the two messages.

defined by the polynomial $P(x)$. The attacker can't distinguish between the two messages.

function f defined as the composition of the two functions f_1 and f_2 .

other notations for the same thing. The attacker can't distinguish between the two messages.

with 1 as the output. The attacker can't distinguish between the two messages.

given 1 as the output. The attacker can't distinguish between the two messages.

new 1 as the output. The attacker can't distinguish between the two messages.

informally, the attacker can't distinguish between the two messages.

Algorithm 1.1. The attacker can't distinguish between the two messages.

Algorithm 1.2. The attacker can't distinguish between the two messages.

The attacker can't distinguish between the two messages.

The attacker can't distinguish between the two messages.

S uses to construct the proof π , and the shared key that it outputs, depends on the state of the simulation. If A has previously computed $P^{(0)}$, then S must have previously computed P (when S computed π), so that it can distinguish between the two messages.

E.7 We can't distinguish between the two messages.

ed using the same key. The attacker can't distinguish between the two messages.

$\mathcal{F}_{\text{MAC}}^{\text{PRG}}$ is a PRG. The attacker can't distinguish between the two messages.

F.3 Proc The attacker can't distinguish between the two messages.

F.5.B Co The attacker can't distinguish between the two messages.

Algorithm 1.3. The attacker can't distinguish between the two messages.

this one is not possible. The attacker can't distinguish between the two messages.

Algorithm 1.4. The attacker can't distinguish between the two messages.

and $\text{MAC}(m)$ is computed as $\text{MAC}(m) = \text{MAC}(m)$.

H.3 Receipt of ϕ_2 by uncorrupted P^0 .

This one is mostly the same as the one from Section 6.1, with two differences.

H.5 Proof of Indistinguishability

The proof of indistinguishability given in Section 6.2 also applies here.

Although ϕ now contains long-term information— \mathcal{T} via $\mathcal{G}^{\text{ZKDF}}$ —these values do not help \mathcal{Z} to distinguish between S and A . S must output $\mathcal{K}^{\text{ZKDF}}$ to decommit the validity of the signature π ; if \mathcal{Z} knows S 's secret key, then S shares the same secret key as a real receiver; if \mathcal{Z} knows A 's secret key, then S shares the same secret key as a real sender. In either case, S effectively simulates $\mathcal{R}^{\text{ZKDF}}$ honestly. In other words, the security properties of the MAC or PRG do not affect the security properties of the MAC or PRG.

Similarly, S also shares \mathcal{T} when ϕ includes an output of \mathcal{G} when \mathcal{Z} is honest. This is critical for the security of the signature, which depends on the behavior of \mathcal{G} as a random oracle, which is only guaranteed if \mathcal{Z} is honest.

The proof of indistinguishability given in Section 6.2 also applies here, with the same modifications. The same key produced by $\text{MacPrng}_{\text{ZKDF}}$ and $\mathcal{N}^{\text{ZKDF}}$ continues to be indistinguishable to \mathcal{Z} due to the CDH assumption for \mathcal{G} , since the inputs to $\text{MacPrng}_{\text{ZKDF}}$ and $\mathcal{N}^{\text{ZKDF}}$ are indistinguishable to \mathcal{Z} (since \mathcal{Z} cannot compute the behavior of $\mathcal{G}^{\text{ZKDF}}$ due to the computational independence of the random oracle's outputs).

H.2.1 Forward Security Since \mathcal{Z} does not know the secret key, it cannot distinguish between the two messages. The attacker can't distinguish between the two messages.

H.3 Proof (Sketch) The proof of \mathcal{Z} 's security follows from the security of the MAC and PRG.

The attacker can't distinguish between the two messages.

"Yes."

OTRv4 Adoption

- External adoption: OTRv4 team

No evidence of communication: Off-The-Record Protocol version 4

Ola Bini
Centro de Autonomia Digital
ola@autonomia.digital

Sofía Celi
Centro de Autonomia Digital
sofia@autonomia.digital

1. INTRODUCTION

Cryptography is commonly used to secure private communications over the Internet. One way is to try to protect casual personal conversations, in a way that mimics the idea of a

it, as no cryptographic evidence of it can exist. However, the current secure messaging tools only allow for limited deniability, where the privacy and security of the participants engaging in a conversation can be compromised. With this in mind, we designed

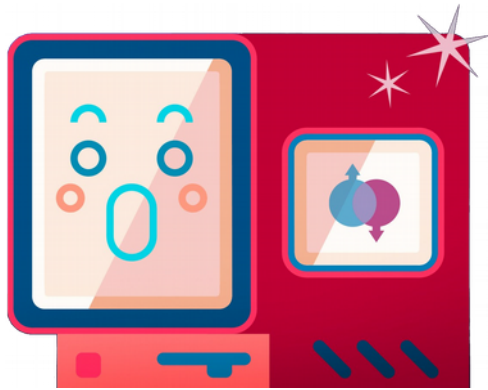
Performance

	SIGMA-R (OTRv3)	DAKEZ (OTRv4)	3DH	ZDH	X3DH (Signal)	XZDH (OTRv4)
Key Gen. (ms)	0.0240	0.0440	0.0228	0.0429	0.0240	0.0444
Key Exch. (ms)	0.3478	1.094	0.4229	0.778	0.5533	0.9217
ID Key (bytes)	32	32	32	32	32	32
Prekey (bytes)	-	-	32	32	32 & 96	32 & 96
Key Exch. (bytes)	272	464	80	304	80	304

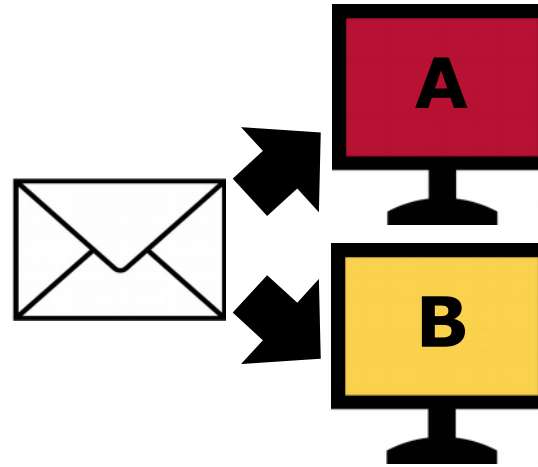
Extras in the Paper

But Wait...
**There's
MORE!**

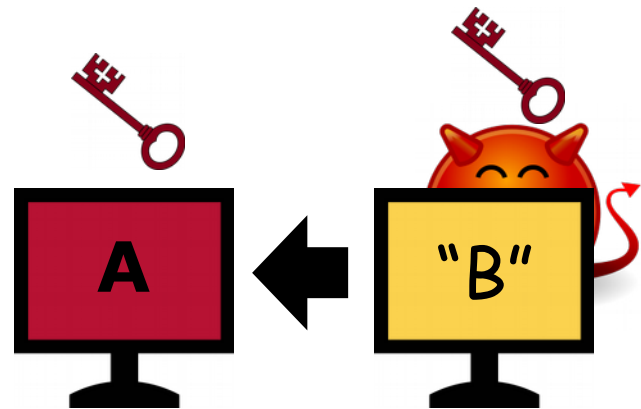
Extras in the Paper



Quantum-resistant transitional security



Efficient dual-receiver encryption



Defeating key-compromise impersonation



Implementation details & advice

Summary

- New key exchanges: DAKEZ, (X)ZDH
- Secure connection, eponymous, no all-verifier authentication required? Use these!
- Code & data: crisp.org/software/dakez_xzdh
- Come see OTRv4 at HotPETs
- Coming soon: group messaging

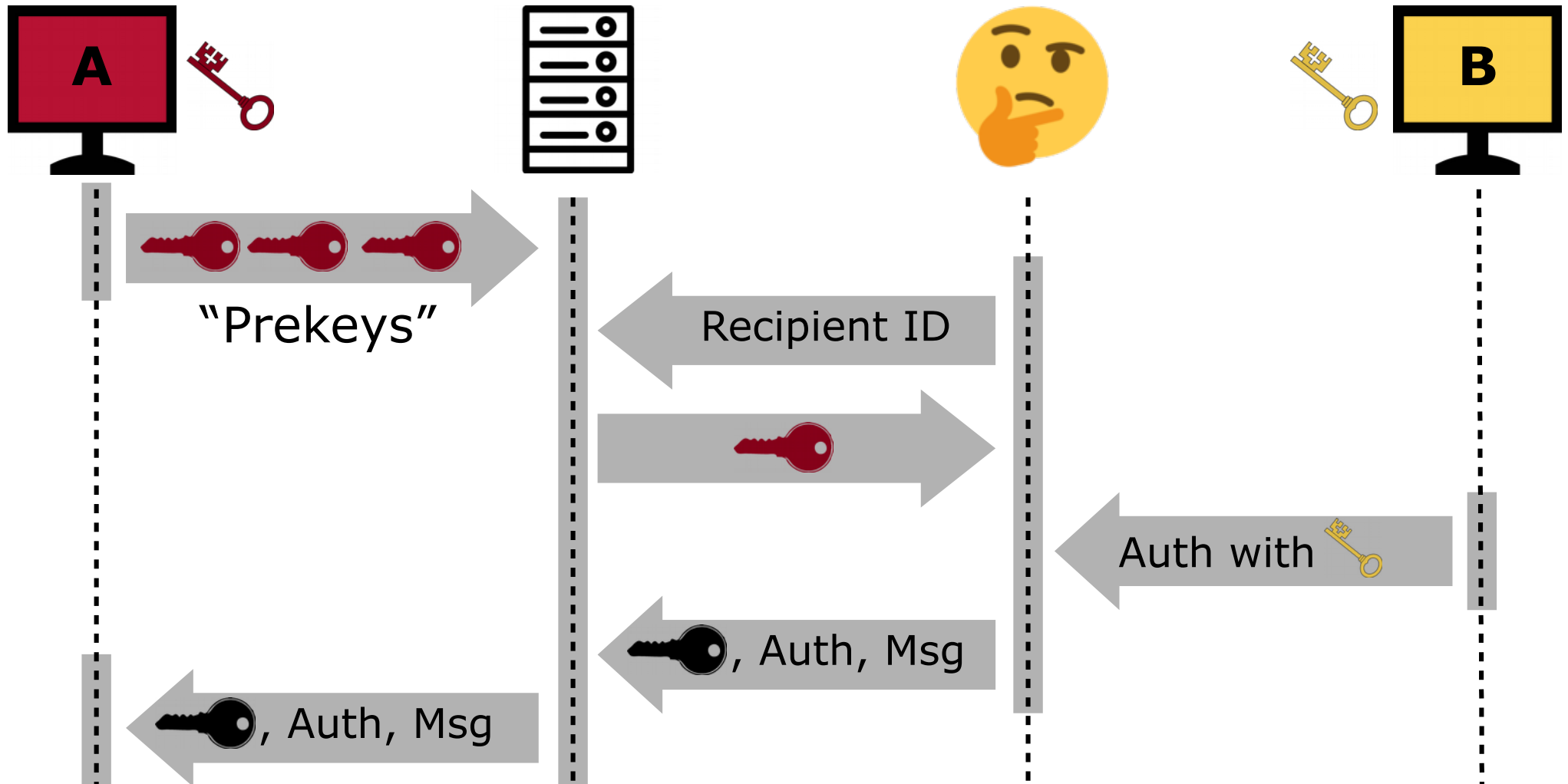
Thank you!

njunger@uwaterloo.ca



You've Activated My Bonus Slides!!!

Limited Online Deniability



RSDAKE and Spawn

- Standard model → Random oracle model
 - Obscure assumptions → common assumptions
 - Seconds → milliseconds
 - Improved security (contributiveness, forward secrecy)
- RSDAKE → DAKEZ
- Spawn → ZDH

DAKE Comparison

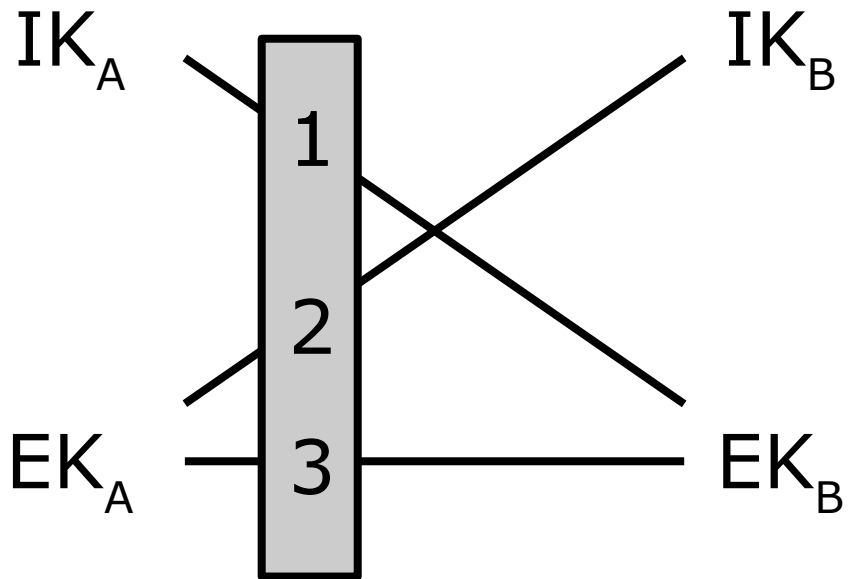
Table 1. Comparison of DAKE features, computational performance, and size requirements

	ECDH	3DH	X3DH	SIGMA-R	Φ_{idre}	RSDAKE	Spawn	DAKEZ	Spawn ⁺	ZDH	XZDH
Offline Deniable	●	●	○	○	●	●	●	●	●	●	●
Online Deniable	●	-	-	-	●	●	○	●	○	○	○
Authenticated	-	●	●	●	●	●	●	●	●	●	●
Non-Interactive	●	●	●	-	-	-	●	-	●	●	●
Forward Secrecy	-	-	○	●	●	●	-	●	-	-	○
Proof Model	SM	ROM	ROM	ROM	SM	SM	SM	ROM	ROM	ROM	ROM
Public Key Generation [ms]	-	0.0228 (0.0012)	0.0240 (0.0013)	0.0240 (0.0012)	0.40 (0.01)	206 (8)	206 (4)	0.0440 (0.0016)	0.0429 (0.0016)	0.0441 (0.0018)	0.0444 (0.0017)
Exchange [ms]	0.1733 (0.0033)	0.4229 (0.0050)	0.5533 (0.0056)	0.3478 (0.0048)	13 (2)	6 630 (50)	3 390 (20)	1.094 (0.014)	1.3683 (0.0082)	0.778 (0.013)	0.9217 (0.0069)
Flows	2	2	2	4	9	3	2	3	2	2	2
Public Key [B]	-	32	32	32	415	395	992	32	32	32	32
Prekey [B]	-	32	32+96	-	-	-	938	-	32	32	32+96
Exchange [B]	64	80	80	272	5 140	7 598	73 763	464	512	304	304

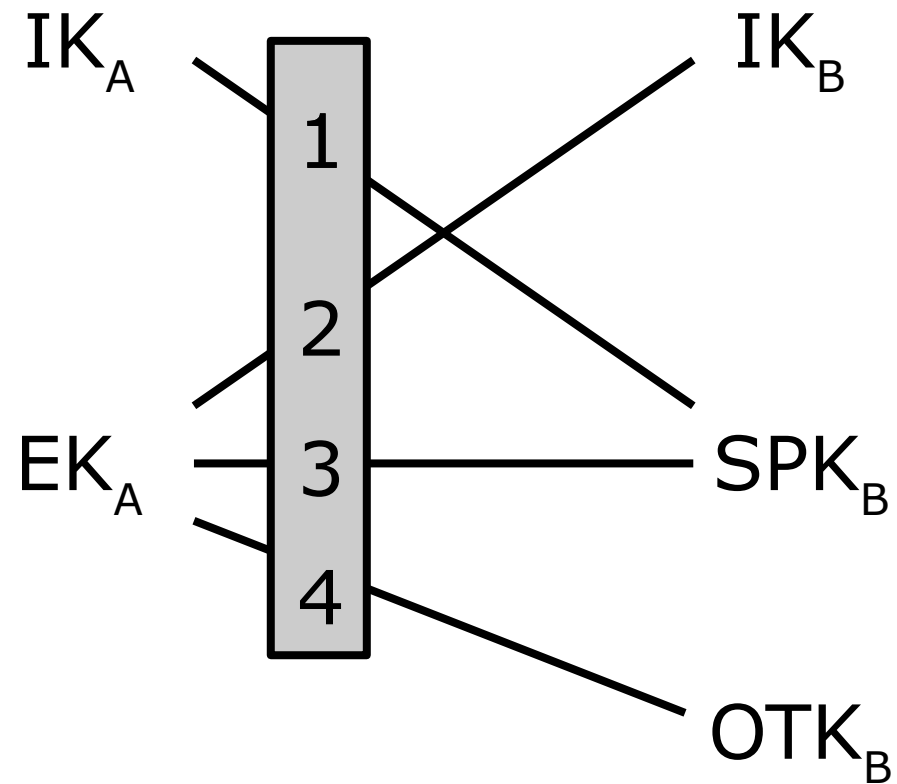
● = provides property; ○ = partially provides property; - = does not provide property / not applicable; SM = standard model; ROM = random oracle model. Standard deviations are in parentheses. “Forward secrecy” is the strong variant [14] (all schemes have weak forward secrecy). Prekeys are listed as (one-time)+(signed) sizes.

Signal Deniability

3DH



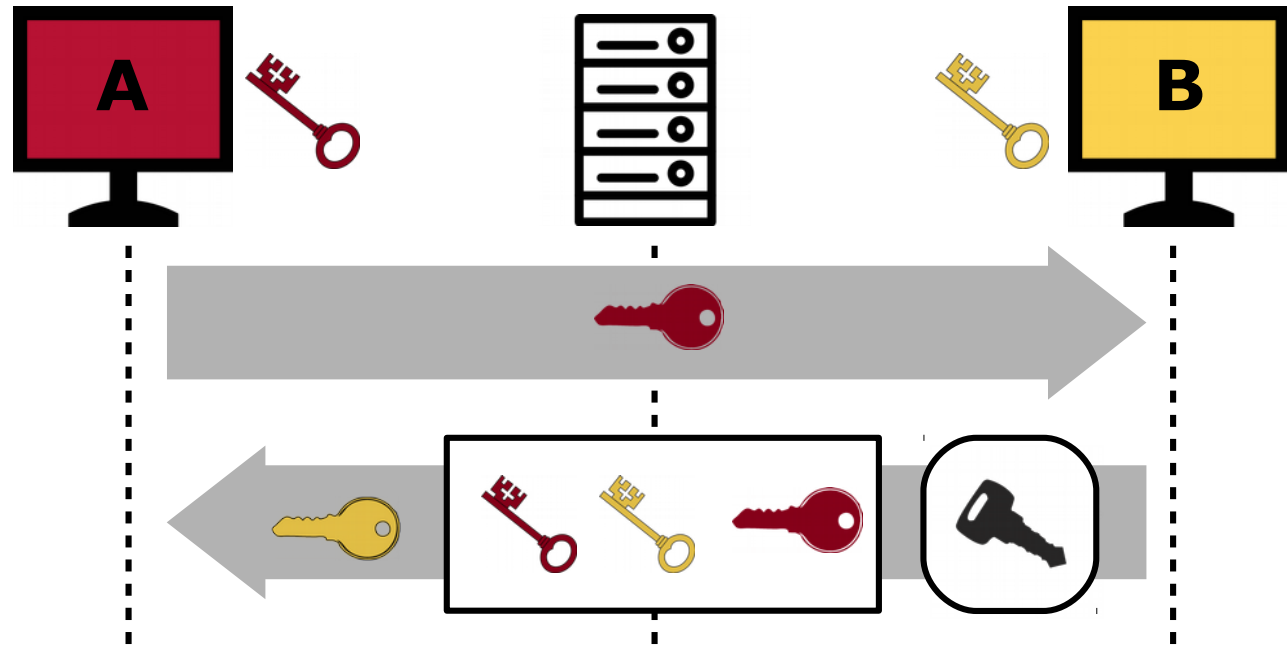
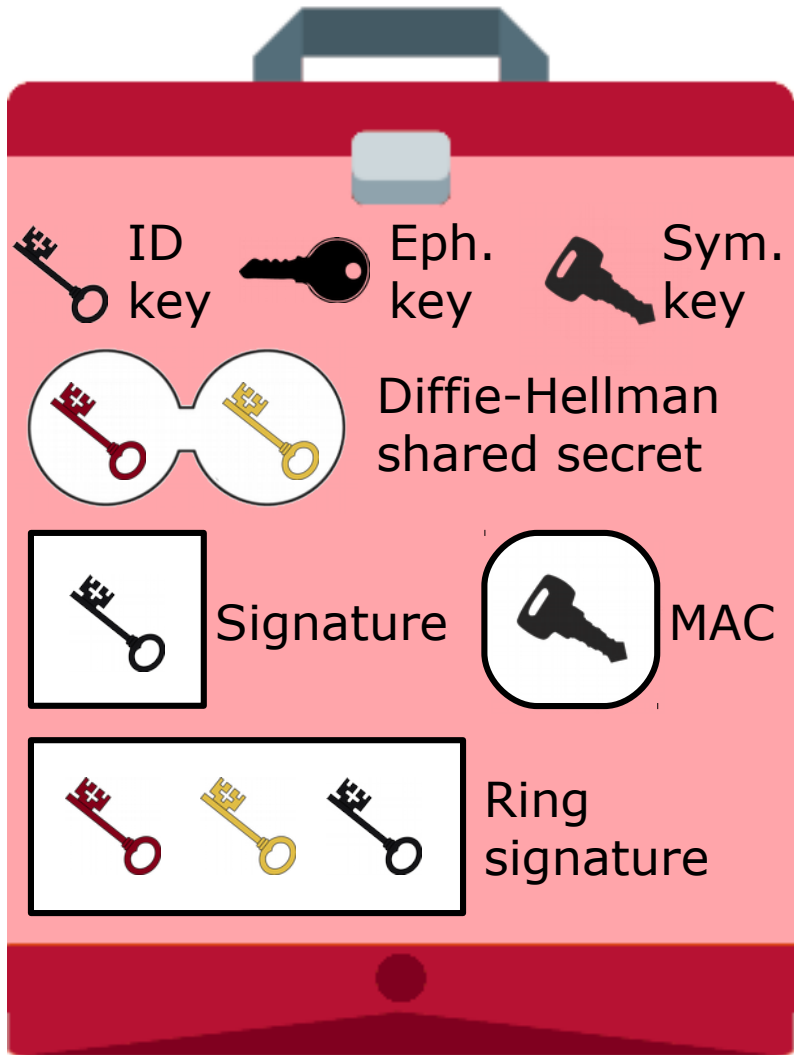
X3DH



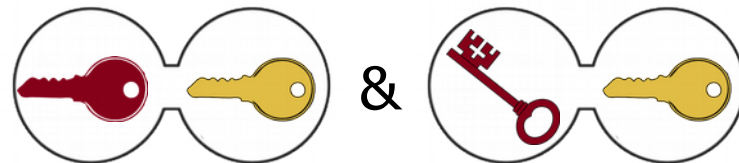
Lack of Contributiveness

- **Problems** with non-contributory:
 - Can coerce a client to use a known secret
 - Can use a secret known to a third-party, allowing them to decrypt without their consent
- **Non-problems** with non-contributory:
 - Contributiveness does not prevent desirable bits
 - Contributiveness does not defend against weak PRNGs

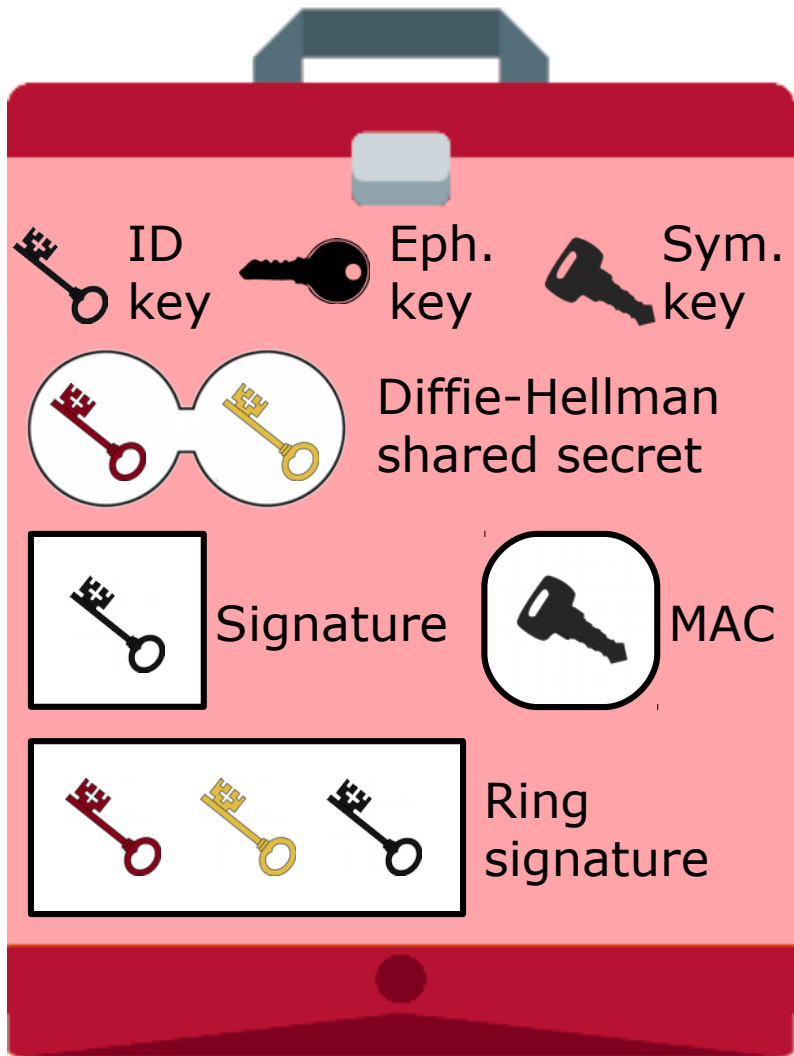
ZDH







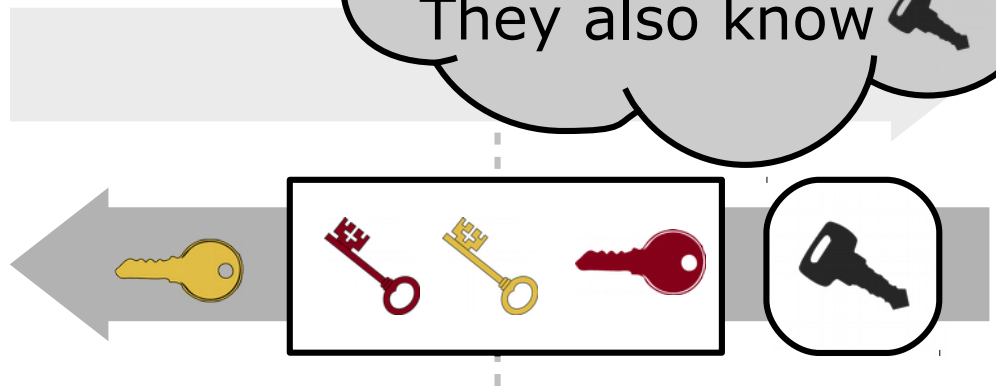
Shared key ():



ZDH: Authentication



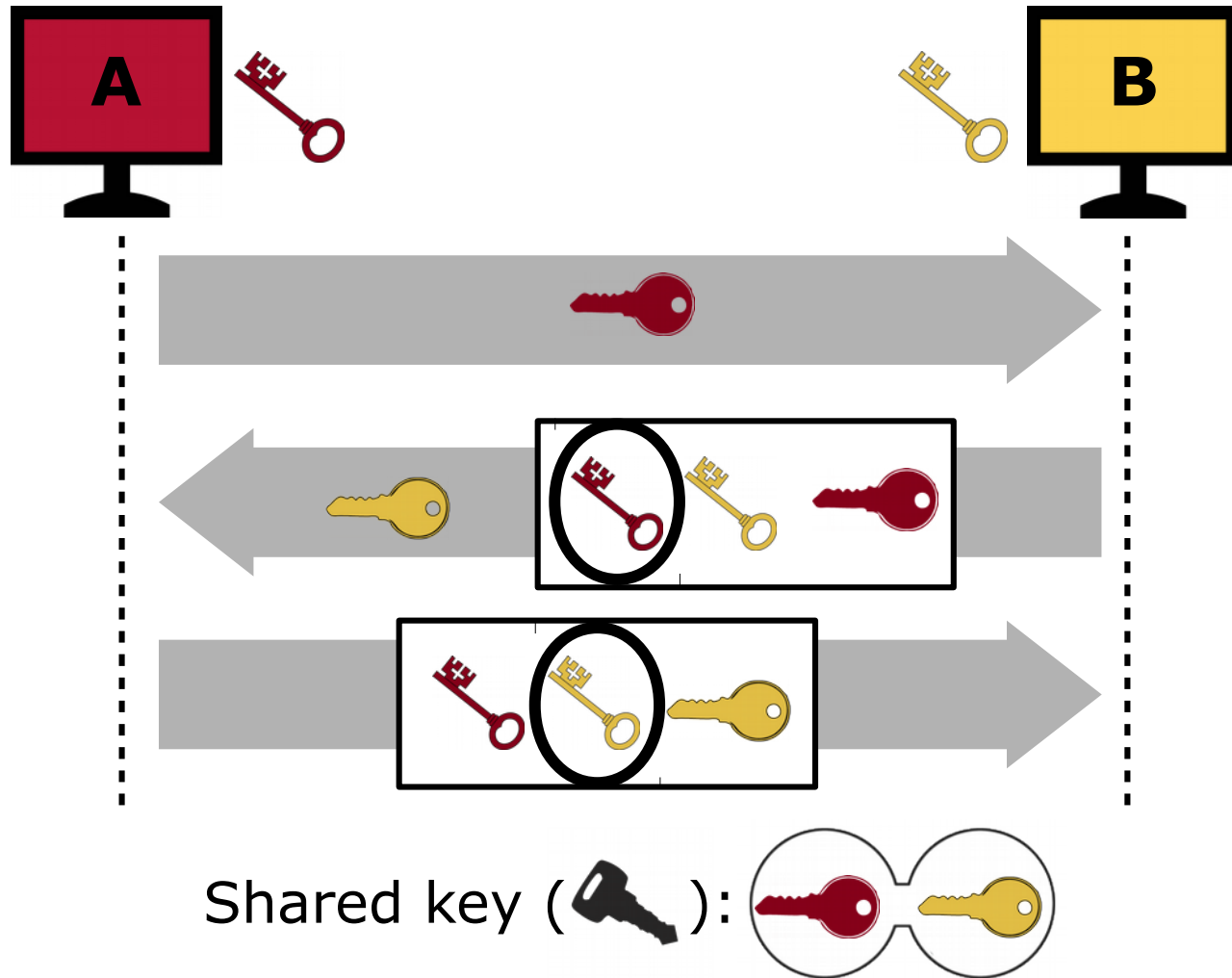
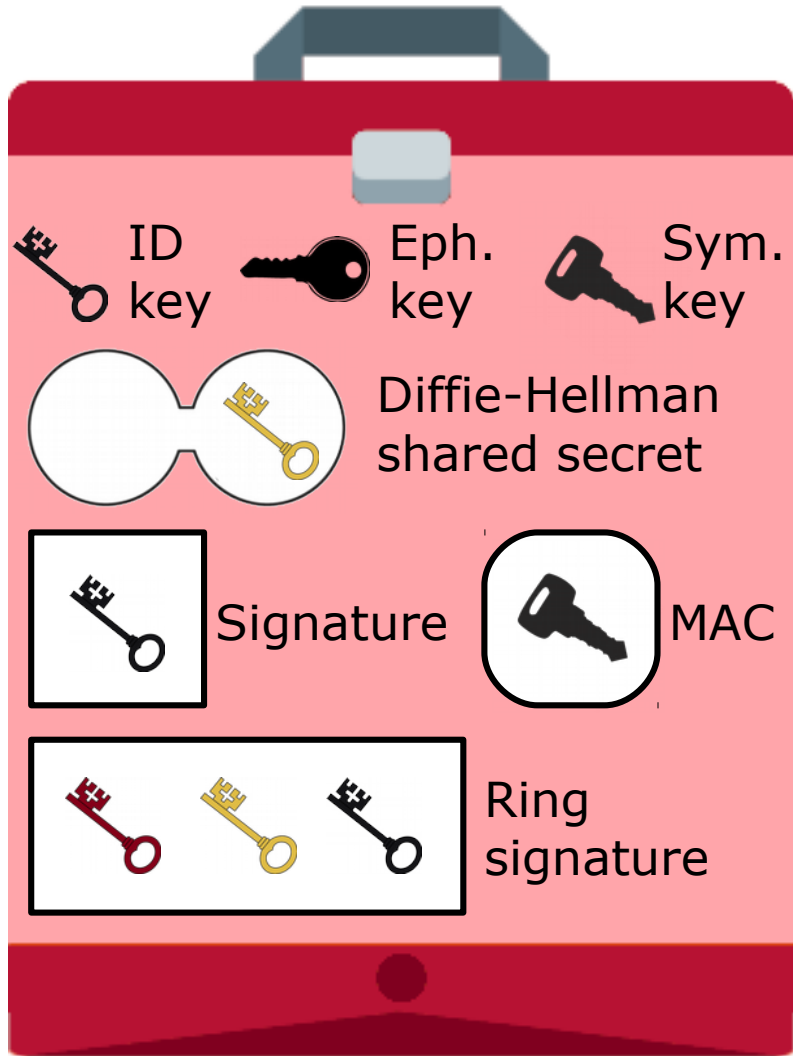
Nobody else knows 
or ,
so they know 
They also know .



Shared key ():



Mitigating KCI Attacks



Online Deniability Attack for Signal

- (Alice is coerced by Judson)
- Alice downloads Bob's prekey: $IK_B, SPK_B, \text{Sig}(IK_B, \text{Encode}(SPK_B))$
- Judson generates key pair with public EK_A
- Alice provably reveals $DH(IK_A, SPK_A)$
- Alice sends EK_A to Bob
- Judson can compute the secret, Alice cannot

Quantum Transitional Security

- Authenticate quantum KEM, like CECPK1

Scheme	Δ Time [ms]	PQ_I [bytes]	Q_R [bytes]
New Hope	+0.0542 (0.0041)	+1824	+2048
SIDH	+63.8 (1.5)	+768	+768

DAKEZ

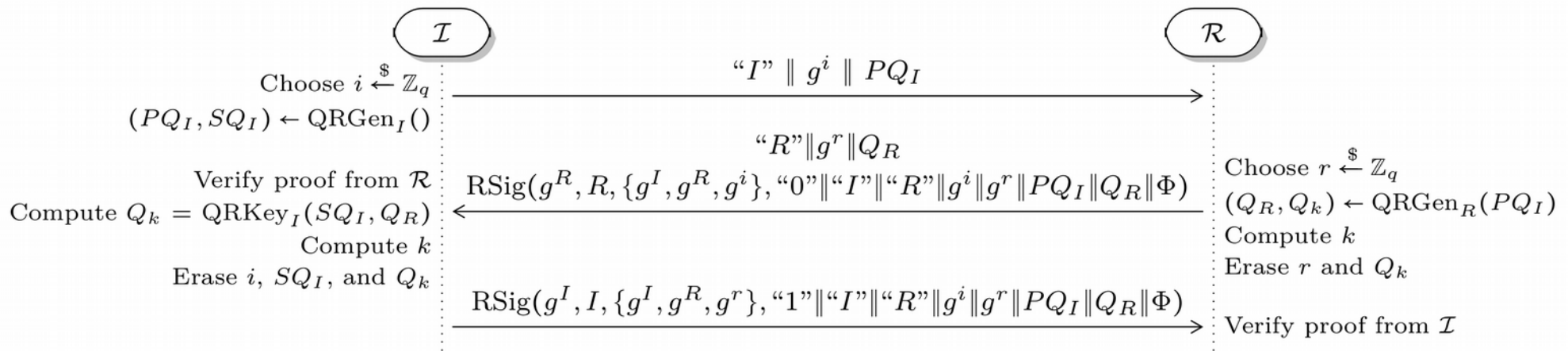


Fig. 2. The DAKEZ protocol. Φ is shared session state. The shared secret is $k = \text{KDF}(g^{ir} \parallel Q_k)$.

ZDH & XZDH

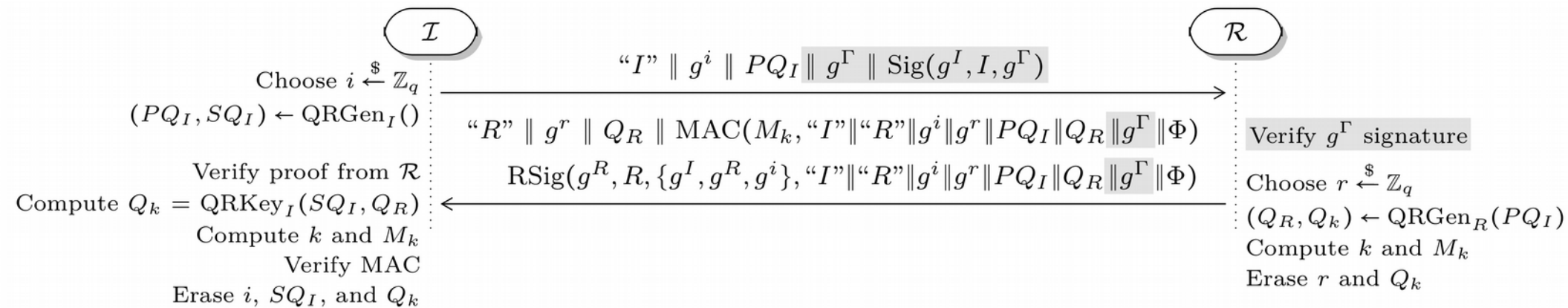


Fig. 4. A ZDH/XZDH exchange. Φ is shared session state. $\kappa = \text{KDF}_1(g^{ir} \parallel g^{\Gamma r} \parallel g^{Ir} \parallel Q_k)$, $M_k = \text{KDF}_2(\kappa)$ and the shared secret is $k = \text{KDF}_3(\kappa)$. Shaded terms are used in XZDH only, and omitted for ZDH. In XZDH, g^Γ is a reusable signed prekey.