No evidence of communication and morality in protocols: Off-the-Record protocol version 4 (OTRv4)

Sofía Celi, Jurre van Bergen

CENTRO DE AUTONOMIA DIGITAL

Why we need secure messaging

“Most academic cryptographers seem to think that our field is a fun, deep, and politically neutral game -a set of puzzles involving communicating parties and notional adversaries. This vision of who we are animates a field whose work is intellectually impressive and rapidly produced, but also quite inbred and divorced from real-world concerns. Is this what cryptography should be like? Is it how we should expend the bulk of our intellectual capital?”

-Rogaway, P. (2015), *The Moral Character of Cryptographic Work*, University of California, Davis, USA

“An especially problematic excision of the political is the marginalization within the cryptographic community of the secure-messaging problem, an instance of which was the problem addressed by Chaum. Secure-messaging is the most fundamental privacy problem in cryptography: how can parties communicate in such a way that nobody knows who said what. More than a decade after the problem was introduced, Racko and Simon would comment on the near-absence of attention being paid to the it“

-Rogaway, P. (2015), The Moral Character of Cryptographic Work,
University of California, Davis, USA

Why we need protocols

- We need options that work
- We need full specifications
- We need properties, limitations and requirements
- We need protocols that update existing definitions: vague terms get better defined
- We need reviews and verifications
- We need ideas from different places
- We need implementations

What is OTR and what is deniability?

In the beginning..

- Paper in 2004 by *Ian Goldberg*, *Nikita Borisov* and *Eric Brewer*
- Conversations in the "digital" world should mimic casual real world conversations
- Authentication in a deniable way
- Introduces the Socialist Millionaires Protocol in OTRv2
- OTR gave inspiration to other secure messaging protocols, like Signal

Off-The-Record

- Authentication
  - As AKE, it uses a variant of the SIGMA protocol
- Verification
  - Socialist millionaire protocol
  - Fingerprint comparison
- End-to-end encryption
  - All messages are encrypted

Off-The-Record

- Perfect Forward secrecy:
  - Usage of unique keys for the encryption of each message
  - "The idea of perfect forward secrecy (sometimes called break-backward protection) is that previous traffic is locked securely in the past." (Menezes, A., Oorschot, P., Vanstone, S. (1997), *Handbook of Applied Cryptography*, CRC Pres.)
  - "A classical adversary that compromises the long-term secret keys of both parties cannot retroactively compromise past session keys" (Bellare, M., Pointcheval, D., & Rogaway, P. (2000). *Authenticated Key Exchange Secure Against Dictionary Attacks*. In Advances in Cryptology–EUROCRYPT)

Off-The-Record

- Post-compromise security (sometimes referred as backward secrecy):
  - Even if a message key gets compromised, no future messages can be decrypted
  - “A protocol between Alice and Bob provides Post-Compromise Security (PCS) if Alice has a security guarantee about communication with Bob, even if Bob’s secrets have already been compromised” (Cohn-Gordon, K., Cremers, C., & Garrat, L. (2016). *On Post-Compromise Security*. Department of Computer Science, University of Oxford)

Deniability

What is deniability?

- “Deniability, also called repudiability, is a common goal for secure messaging systems. Consider a scenario where Bob accuses Alice of sending a specific message. Justin, a judge, must decide whether or not he believes that Alice actually did so. If Bob can provide evidence that Alice sent that message, such as a valid cryptographic signature of the message under Alice’s long-term key, then we say that the action is non repudiable. Otherwise, the action is deniable”
- Unger, N., Dechand, S., Bonneau, J., Fahl, S., Perl, H., Goldberg, I., Smith, M. (2015), *SoK: Secure Messaging*, 2015 IEEE Symposium on Security and Privacy

Types

- Online, offline, message, participation
  “We can distinguish between message repudiation, in which Alice denies sending a specific message, and participation repudiation in which Alice denies communicating with Bob at all.”
  - Unger, N., Dechand, S., Bonneau, J., Fahl, S., Perl, H., Goldberg, I., Smith, M. (2015), *SoK: Secure Messaging*, 2015 IEEE Symposium on Security and Privacy

Types

"A protocol is strongly deniable if transcripts provide no evidence even if long-term key material is compromised (offline deniability) and no outsider can obtain evidence even if an insider interactively colludes with them (online deniability)."

- Unger, N. & Goldberg, I. (2015), *Improved Strongly Deniable Authenticated Key Exchanges for Secure Messaging*, University of Waterloo, Waterloo, Canada.

	OTRv3	OTRv4	Signal	OMEMO	Olm/Megolm	Telegram
Forward secrecy	Weak	Interactive: full Non- interactive: weak	Weak	Weak	None	Weak*
Post-compromise secrecy	Full	Full	Full	Full	Full	Full*
Online Deniability	○	● ◐	○	○	○	○
Offline Deniability	◐	●	◐	◐	●	●

●	provides property
◐	partially provides property
○	does not provide property

Why a version 4 of OTR?

- We want deniability: participation, message, online and offline
- We want perfect forward and post-compromise secrecy
- We want a higher security level
- We want to update the cryptographic primitives
- We want additional protection against transcript decryption in the case of ECC compromise
- We want elliptic curves

New communication model

- We want in-order and out-of-order delivery of messages
- We want online and offline conversations
- We want different modes in which something can be implemented
- We don’t want to trust servers

Main Changes over Version 3

- Security level raised to 224 bits and based on Elliptic Curve Cryptography (ECC).
- Additional protection against transcript decryption in the case of ECC compromise.
- Support of conversations where one party is offline.
- The cryptographic primitives and protocols have been updated:
  - Deniable authenticated key exchanges (DAKE) using "DAKE with Zero Knowledge" (DAKEZ) and "Extended Zero-knowledge Diffie-Hellman" (XZDH) [1]. DAKEZ corresponds to conversations when both parties are online (interactive) and XZDH to conversations when one of the parties is offline (non-interactive).
  - Key management using the Double Ratchet Algorithm [2].
  - Upgraded SHA-1 and SHA-2 to SHAKE-256.
  - Switched from AES to XSalsa20 [3].
- Support of an out-of-order network model.
- Support of different modes in which this specification can be implemented.
- Explicit instructions for producing forged transcripts using the same functions used to conduct honest conversations.

Design

- Why DAKEZ/XZDH instead of something simpler?
- Why Ed448-Goldilocks?
- Why DH-3072?
- Why SHAKE? Why XSalsa20?
- Usage of the Double Ratchet Algorithm
- What is the toolkit?
- Why not post-quantum algorithms?
- Why no group chat?

Real world implementation

Applied cryptography

- Collaboration with cryptographers and developers:
  - libgoldilocks as an extension of libdecaf from Mike Hamburg
  - Java, python and golang implementations
  - Collaboration with cryptographers while they were writing papers
  - Revisions by Ian Goldberg and Nik Unger

Nik Unger and Ian Goldberg

Improved Strongly Deniable Authenticated Key Exchanges for Secure Messaging*

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

Ed448-Goldilocks

A 448-bit Edwards curve

Brought to you by: bitwiseshiftlef

Implementation in C

- Why in C?
- C memory handling: how to check it?
- Testing: unit and integration
- Static testing: clang-tidy and splint
- Valgrind and various sanitizers

- Code that can be readable
- Code that can be used by other developers
- Recommendations to developers
- In touch with the community

Testing on various systems

- Why it is important to test in multiple OS: older versions of Linux
- BSD's
- Running the test suite on exotic architectures

UI/UX work // Formal verifications

- The user matters
- Make dialogs more understandable

- Model checkers
- Testing the protocol state machine in C-Murphy
- Eventually, we want a full protocol formal verification

Security audits

- Introducing fuzzing: Libfuzzer and OSS-Fuzz
- We welcome community audits
- We will get a security audit

Check out our repos!

The protocols:

https://github.com/otrv4/otrv4

https://github.com/otrv4/otrv4-prekey-server

The library:

https://github.com/otrv4/libotr-ng

The plugin:

https://github.com/otrv4/pidgin-otrng

The prekey server:

- https://github.com/otrv4/otrng-prekey-server
- https://github.com/otrv4/prekey-server-xmpp

The toolkit:

- https://github.com/otrv4/libotr-ng-toolkit

Golang

https://github.com/otrv4/otr4

Java by Danny van Heumen

https://gitlab.com/cobratbq/otr4j

OTR.im

- Happy to host you and setup CI/CD

Thanks to everyone involved

To the main collaborators (people in the current team or with more than 6000 lines of code/text contributed):

- Ian Goldberg
- Nik Unger
- Mike Hamburg
- Sofia Celi
- Ola Bini
- Reinaldo de Souza Jr
- Rosalie Tolentino
- Jurre van Bergen
- Iván Pazmiño
- Giovane Liberato
- Fan Jiang
- Others who have collaborated

Time for references

1. Goldberg, I. and Unger, N. (2016). Improved Strongly Deniable Authenticated Key Exchanges for Secure Messaging, Waterloo, Canada: University of Waterloo. Available at: http://cacr.uwaterloo.ca/techreports/2016/cacr2016-06.pdf

2. Hamburg, M. (2015). Ed448-Goldilocks, a new elliptic curve, NIST ECC workshop. Available at: https://eprint.iacr.org/2015/625.pdf

3. Gunn, L. J., Vieitez Parra, R. and Asokan, N. (2018) On The Use of Remote Attestation to Break and Repair Deniability. Available at: https://eprint.iacr.org/2018/424.pdf

4. Rogaway, P. (2015), *The Moral Character of Cryptographic Work*, University of California, Davis, USA

5. Menezes, A., Oorschot, P., Vanstone, S. (1997), *Handbook of Applied Cryptography*, CRC Pres.)

6. Bellare, M., Pointcheval, D., & Rogaway, P. (2000). *Authenticated Key Exchange Secure Against Dictionary Attacks*. In Advances in Cryptology–EUROCRYPT

7. Cohn-Gordon, K., Cremers, C., & Garrat, L. (2016). *On Post-Compromise Security*. Department of Computer Science, University of Oxford

8. Unger, N., Dechand, S., Bonneau, J., Fahl, S., Perl, H., Goldberg, I., Smith, M. (2015), *SoK: Secure Messaging*, 2015 IEEE Symposium on Security and Privacy

Questions?

- Come find us at the Off-The-Record assembly!

- Or online! https://otr.im/
- IRC: #otr at OFTC

Thanks!

Jurre van Bergen
@DrWhax

Sofía Celi
@cherenkov_d

You have unlocked the secret slides*

*Copyright to Nik Unger

Difference with OMEMO

- OTRv4 is agnostic: can work over any protocol, even asynchronous
- OTRv4 has better deniability properties
- OTRv4 has a well defined specification

Difference with Signal

- OTRv4 has better deniability properties and perfect forward secrecy
- OTRv4 has a well defined specification
- OTRv4 has different verification mechanisms
- OTRv4 supports different networks and is not centralized
- OTRv4 supports other features, such as symmetric keys

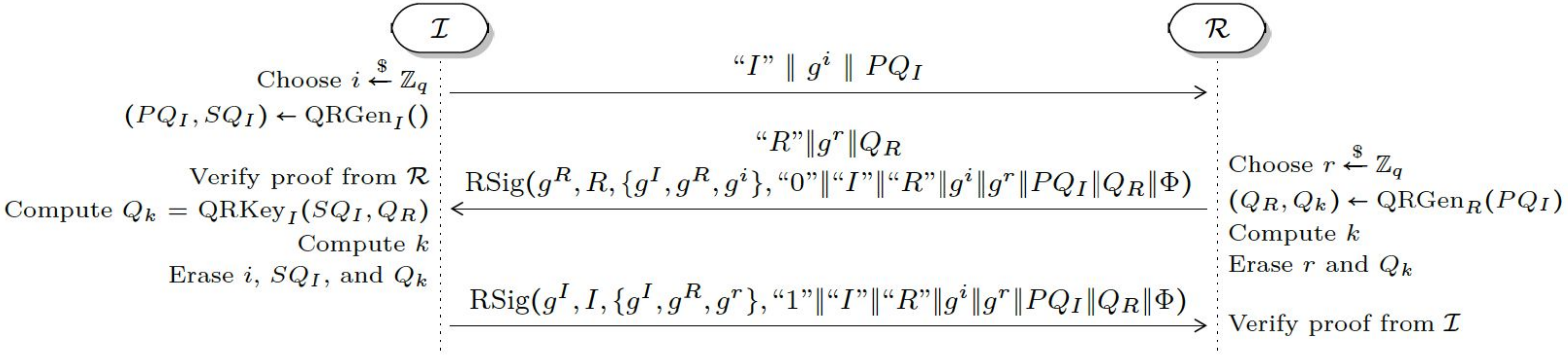
Why deniability matters

- It is a right in casual real-world conversations, even if you don’t think about it
- It is useful not only to you but to whom you are talking to
- It is resistance
- We shouldn’t make the situation worse than plaintext, by adding irrefutable proof of conversations

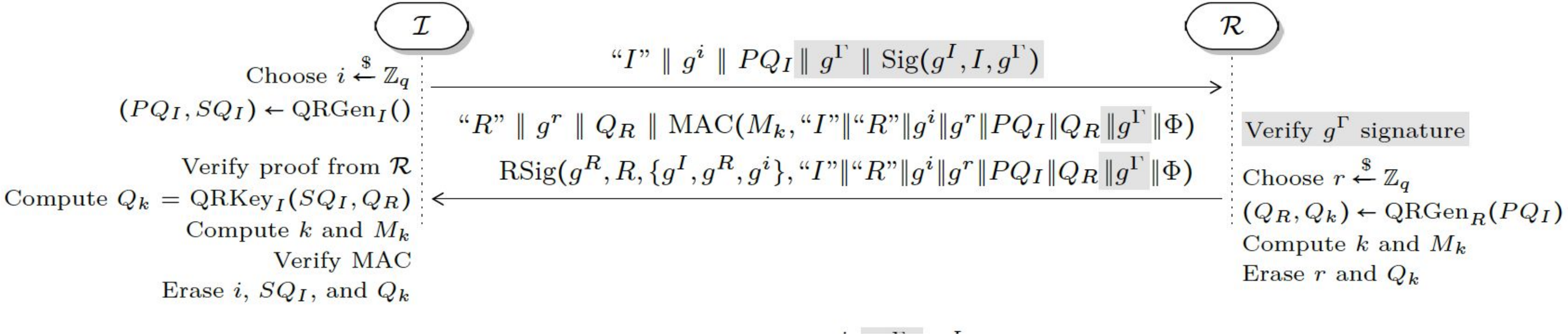
What is weak forward secrecy?

- Strong forward secrecy: protects the session key when at least one party completes the DAKE exchange
- Weak forward secrecy: protects the session key only when both parties complete the DAKE exchange

The DAKEs



DAKEZ -Unger, N. & Goldberg, I. (2015), *Improved Strongly Deniable Authenticated Key Exchanges for Secure Messaging*, University of Waterloo, Waterloo, Canada



XZDH -Unger, N. & Goldberg, I. (2015), *Improved Strongly Deniable Authenticated Key Exchanges for Secure Messaging*, University of Waterloo, Waterloo, Canada