Deniable Key Exchanges for Secure Messaging

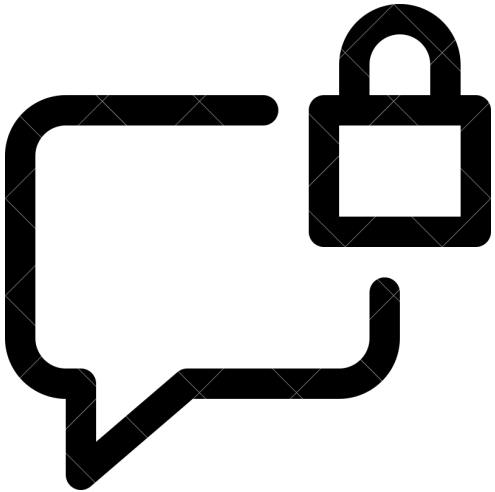
Nik Unger & Ian Goldberg

Presented By: Shubham Agarwal

Motivation

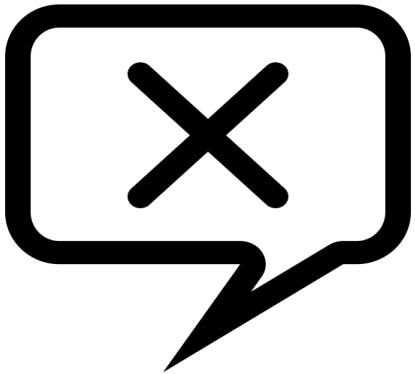
Deniability in Secure Messaging

- What is it?
- What are the existing solutions, if there are any?
- How do we define deniability?
- What do we discuss today?



Deniability/Repudiation

- Plausible Deniability - Lack of convincing proof that an action occurred.
- Repudiation: Message & Participation Repudiation.
- Judgement Metrics: Valid cryptographic proof of the communication/authorship.
- Goal: No additional evidence against the participant, except protocol transcripts.



Online & Offline Judges - I

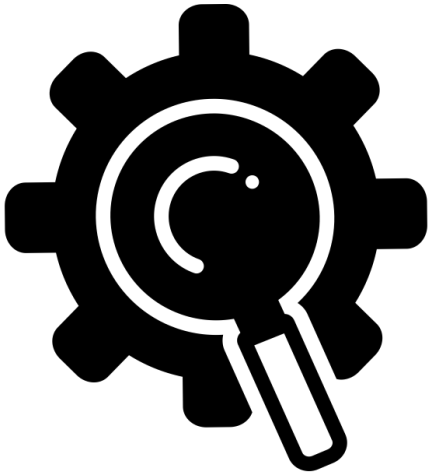
- Consideration: no unforgeable cryptographic proofs as evidence.
- Offline Judge:
  - Evidence: chat transcripts.
  - Assumption: long-term secret keys revealed.
  - **Goal**: prevent distinction between real and fake transcripts.

Online & Offline Judges - II (continued)

- Online Judge:
  - Evidence: Responses provided by informant.
  - Assumption: The judge interacts with informant and instructs them to perform desired actions.
  - **Goal**: The judge decides the actions of informant as real or simulated/fake.

Practicality of Deniability

- Scientific Community:
  - Too expensive to design & implement.
  - Little to no practical relevance or usage.
  - Legal and ethical implications.

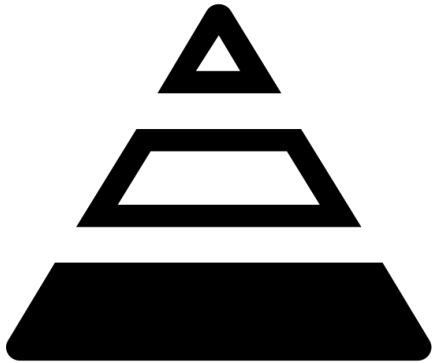


- Author's Opinion:

"we should strive to design deniable protocols to avoid unintentionally incriminating users"

Topics Discussed in this Research Study

- Discussion on ϕ_{dre}, closest known solution, its shortcomings and proposed modifications.
- **RSDAKE** - interactive DAKE which improves the security of ϕ_{dre}.
- **SPAWN** - first non-interactive DAKE; provides forward secrecy and also achieves deniability.
- Proposed extension for *TextSecure* Messaging Application.



Pre-requisites

Cryptographic Constructs - I

Dual-Receiver Encryption - enables publicly verifiable encryption of messages - only either of the two involved entities can read the actual message.

- *DRGen(r)*, *DREnc(pk_1, pk_2, m, r)*, *DRDec(pk_1, pk_2, sk_i, γ)*
- For any *(pk_1, sk_1)* and *(pk_2, sk_2)* produced by *DRGen, for i* $\in$ *{1, 2},* and any *m* and *r*:

  DRDec(pk_1, pk_2, sk_i, DREnc(pk_1, pk_2, m, r)) = m

Cryptographic Constructs - II (continued)

Non-Committing Encryption - Functionalities offered by standard PKE scheme + ability to generate **rigged** ciphertexts.

- *NCGen(r)*, *NCEnc(pk, m, r)*, *NCDec(pk, sk, γ)*, ***NCSim(r)***, ***NCEqv(pk, γ, α, m)***.
- *NCSim(r)* = {*pk*, γ, α} - identically distributed along with outputs of *NCGen* & *NCEnc.*
- *NCEqv(pk, γ, α, m)* = {*sk, r*, r^{NCE}*}; s.t. *NCGen(r*)* = {*pk, sk*}, and *NCEnc(pk, m, r^{NCE}) = γ*.

Cryptographic Constructs - III (continued)

Ring Signatures - digital signature scheme - given a set of *n* members, the ring signature could be verifiably produced by any of them without revealing the exact identity of the signer.

- *RSGen(r)*, *RSig(pk, sk, R, m, r)*, *RVrf(R, σ, m)*
- Ring, *R* - set of *n* public keys - {pk_1, pk_2, ..., pk_n}

Proposed Schemes

The Walfish Protocol, ϕ_{dre} - I

- ϕ_{dre} - only known DAKE which claimed to offer forward secrecy as well as both offline and online repudiation simultaneously.
- Two-round interactive DAKE with non-transferable auth.
- UC Framework extended to **GUC** Framework to prove the security model of the protocol.
- $\mathcal{F}_{ke}$ models the idealized protocol for security guarantees.

Interlude: (G)UC Framework

- **UC Framework** - method to prove that a real protocol behaves identically to an ideal protocol with well-defined security properties.
- It assumes that the protocol in test does not have access to shared information between multiple sessions.
- **GUC Framework** - models the security of multiple concurrent protocol sessions and the shared information exchanges.

The Walfish Protocol, ϕ_{dre} - II (continued)

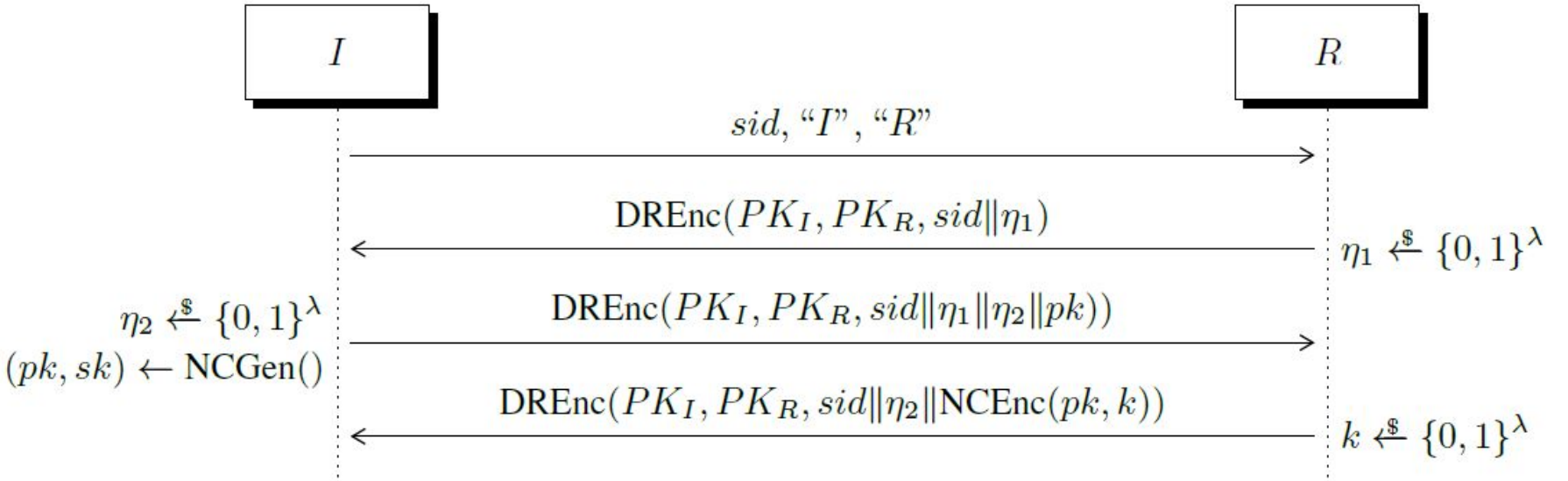


Figure 1: Φ_{dre} [29]. The shared secret is k.

The Walfish Protocol, ϕ_{dre} & IncProc- III (continued)

- Imperfect online deniability in ϕ_{dre} detected.
- Walfish proved that $\mathcal{F}_{ke}$ cannot be realized in the presence of adaptive corruptions.
- **Problem** - What if an adversary, S, disrupts the protocol during its execution such that it aborts?
- **IncProc** - used by the judge to discriminate between a real response and simulated response served by (mis-)informant.

Modified/Proposed ϕ_{idre} - I

- $\mathcal{F}_{keia}$ along with IncProc, models the idealized protocol for security guarantees.
- Non-interactive ZKPK to interactive ZKPK.
- DRE Construction - based on **Cramer-Shoup PKE scheme**.
- **Resulting Protocol** - contains 9 message flows - along with an interactive ZKPK that proceeds between the prover and the verifier.

RSDAKE - I

Φ_{idre} :

- 9 message flows - latency.
- Non-contributory computation of *k*.
- Pre-specified peer KE.

RSDAKE:

- 3 message flows.
- Contributory computation of *k*.
- Post-specified peer KE.

$\mathcal{F}_{post\text{-}keia}$ along with IncProc, models the security guarantees of RSDAKE appropriately.

RSDAKE - II (continued)

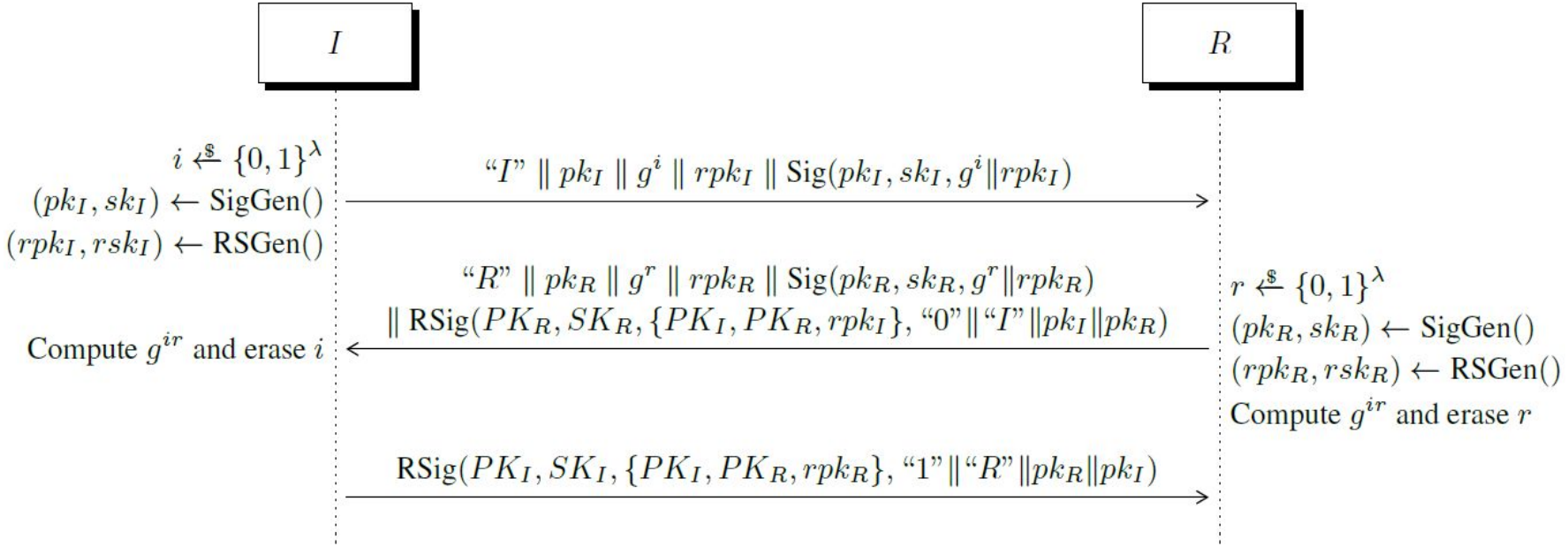


Figure 2: RSDAKE. The shared secret is g^{ir}.

Spawn*- I

- Secure and deniable **one-round key exchange protocol** - suitable for both interactive & non-interactive settings.
- Relies on **central server** - to upload & distribute prekeys.
- Single Post-specified Peer.
- $\mathcal{F}_{1psp\text{-}keia}$ along with **IncProc,** models the security guarantees for the proposed protocol.

Spawn*- II

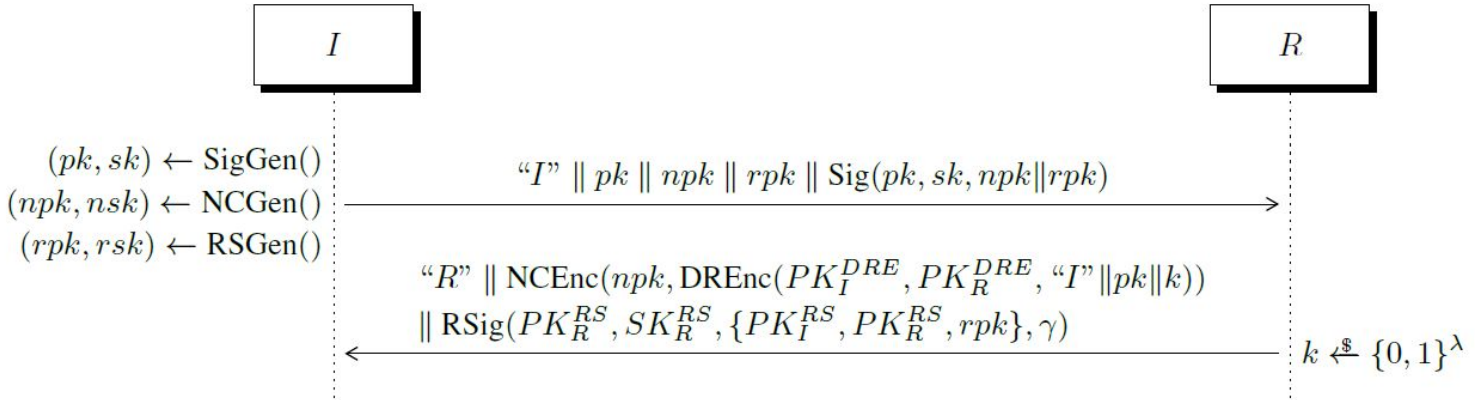


Figure 3: Spawn*. The shared secret is k. γ denotes "R" concatenated with the output of NCEnc. Spawn replaces NCGen with PKGen and NCEnc with PKEnc. In all other ways, Spawn is identical to Spawn*.

Weakness/Limitations

- Incriminating abort by an adversary, *S* - possible, yet under unreasonable settings.
- Security assumption of GUC Framework: if an adversary, *S*, has previously corrupted any party, *P*, it gets access to its SK^{RS}.
- Security assumption of Spawn*: if an adversary, *S*, corrupts any party, *P*, it can get access to *P*'s SK^{DRE}.
- An online judge may discriminate between real and simulated messages when *R* impersonates as *I*.

Spawn* - Summary

- Unlike $\phi_{(i)dre}$ & RSDAKE, Spawn* provides identical security also in non-interactive settings.
- Incriminating abort still possible - but under specific settings.
- Weakened online repudiation.
- Still provides stronger deniability guarantees than 3-DH, the current (non-interactive) KE protocol in *TextSecure*.

TextSecure Iron Triangle - I

- **Conjecture:** Any TextSecure-like one-round KE Protocol **cannot** provide non-interactivity, forward secrecy and online repudiation **simultaneously** - when *R* simulates as *I*.
- Can *R* simulate as *I* to an online judge?
  - Secrets known to *R*: $\{SK_R\}$
  - Secrets missing to recover *k* from transcript: $\{sk_I, sk_R\}$

TextSecure Iron Triangle - II (continued)

- Can *R* deniably simulate sk_I by itself and go unnoticed by an online judge?
- **Problem:** *R* may not have knowledge of sk_I as it could be replaced by genuine sk_I by the online judge. Thus, cannot recover *k* from transcript.
  - If *R* recovers *k* just with SK_R - **no forward secrecy**.
  - If *R* does not recover *k* - no simulation possible - **no online repudiability**

Practicality of Spawn(*)

- (Too) **Strong Threat Model**: adaptive corruption & non-erasure model of encryption.
- Spawn* ⇨ Spawn:
  - either of the above two assumptions are relaxed.
  - NC Encryption replaced by Standard PK Encryption.
- Weaker model/Spawn - practical in real-time environment.

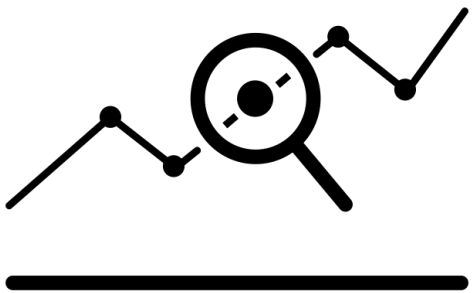
TextSecure & Spawn

TextSecure:

- Key Features: Forward & Backward Secrecy.
- Protocol: 3-DH DAKE
- Protects against offline judges, but not against online judges.

Deniability can be added to TextSecure with Spawn by:

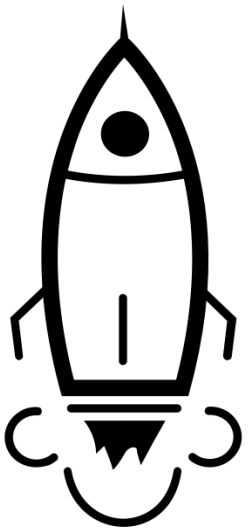
- Replacing 3-DH with Spawn based KE.
- Models the contributory Axolotl by generating new keys from *k*.



Implementation & Evaluation

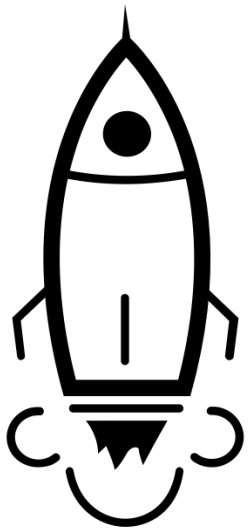
Implementation - I

- Usability Issues with existing solutions.
- Open Implementation to encourage adoption.
- Objective:
  - Implemented $\phi_{dre,}$ ϕ_{idre} , RSDAKE and Spawn.
  - Provably secure in the standard-model.



Implementation - II (continued)

- Along with proposed protocols, the authors implemented few other libraries:
  - Pairing-based Cryptography Wrapper
  - HORS+ Signature Scheme
  - Elliptic Curve Cramer-Shoup Scheme.
  - Chow, Franklin, and Zhang Scheme.
  - Shasham-Waters Scheme.



Performance Evaluation - I

- Simulation: Interactive session between 2 parties over Internet.
- Protocols Evaluated: $\phi_{dre,}$ $\phi_{idre,}$ RSDAKE, Spawn.
- **Metrics**:
  - Security Bits
  - Network Bandwidth
  - Transmission Latency.

Performance Evaluation - II (continued)

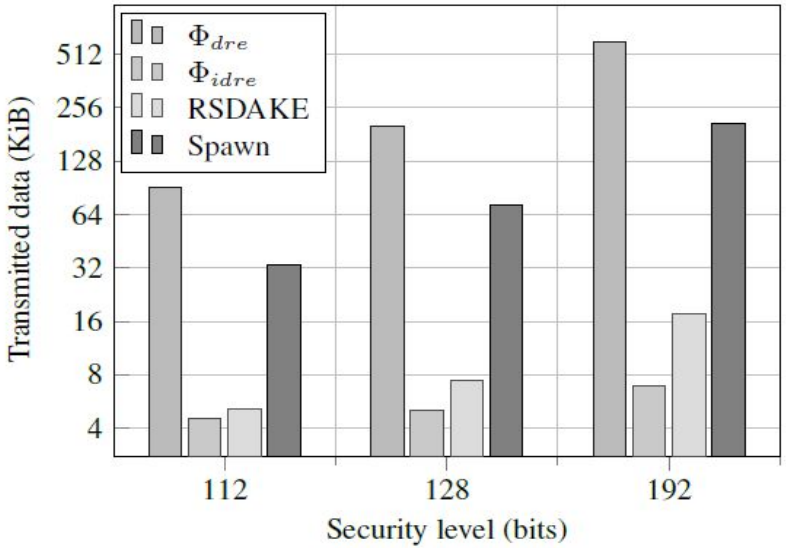


Figure 4: The amount of data transmitted increases significantly with higher security levels. Φ_{dre} and Spawn require significantly more transmissions than Φ_{idre} or RSDAKE.

Performance Evaluation - III (continued)

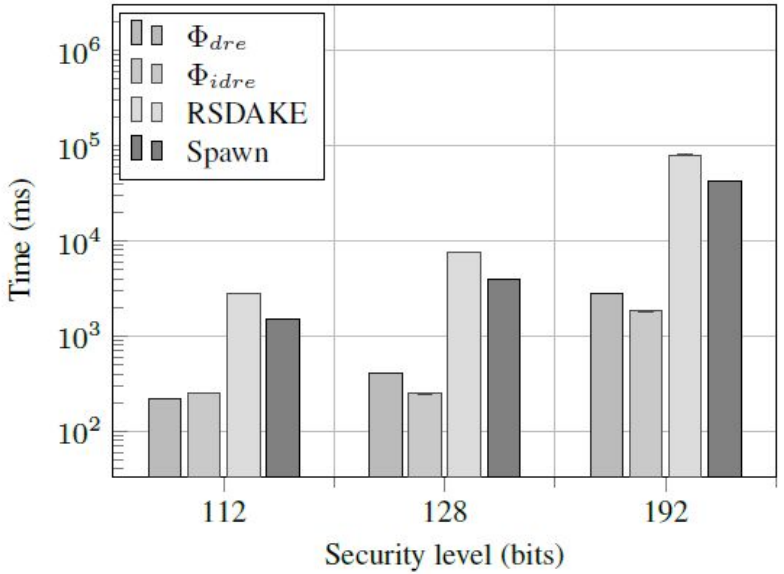


Figure 5: Over a high-bandwidth connection with no latency, the cryptographic overhead of each protocol is clear. The use of ring signatures negatively affects RSDAKE and Spawn.

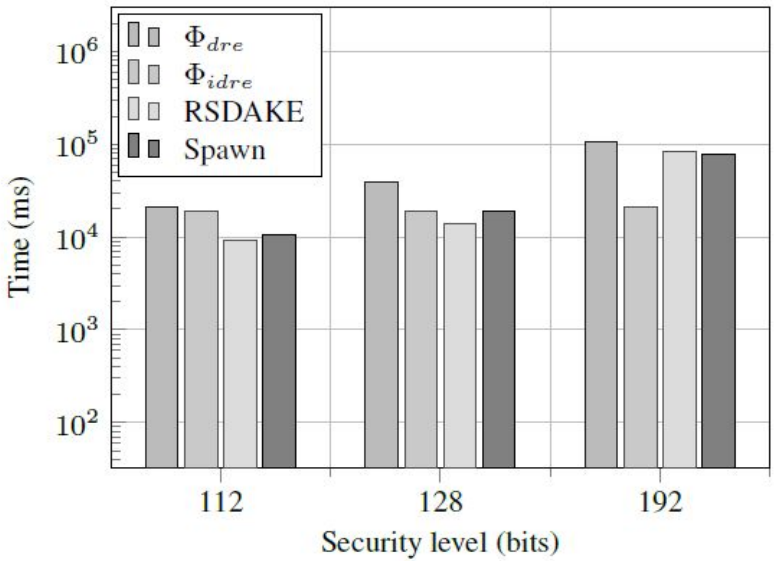


Figure 6: Over a low-bandwidth and high-latency connection, the network significantly affects performance. RSDAKE and Spawn perform the best at 112- and 128-bit security levels.

Performance Evaluation - III (continued)

- In general, all four schemes require increasingly expensive cryptographic operations with increasing security level.
- ϕ_{idre} uses the least data among all four protocols.
- ϕ_{idre} scales well in both high and low latency conditions.
- ϕ_{dre} suffers due to its underlying DRE scheme, in general.
- Performance of RSDAKE and Spawn - hampered by Shacham-Waters scheme.

Conclusion

- **Spawn** - non-interactive protocol with forward secrecy and strong deniability properties.
- **RSDAKE** - interactive substitute for Spawn with additional security property compared to $\phi_{(i)dre}$ - contributory KE.
- ϕ_{idre} **and RSDAKE** - optimal for bandwidth-constrained network. ϕ_{dre} **and** ϕ_{idre} - better suited over large & fast connections.
- Relaxing cryptographic schemes which use random oracle for security may greatly increase the performance of protocols.

Other Relevant Works

- Dodis, Yevgeniy, et al. "**Composability and on-line deniability of authentication.**" Theory of Cryptography Conference. Springer, Berlin, Heidelberg, 2009.
- Unger, Nik, and Ian Goldberg. "I**mproved strongly deniable authenticated key exchanges for secure messaging.**" Proceedings on Privacy Enhancing Technologies 2018.1 (2018): 21-66.
- Tian, Yangguang, et al. "**DABKE: Secure deniable attribute-based key exchange framework.**" Journal of Computer Security Preprint (2019): 1-17.

Potential Future Works

- Agreement on standard definition of deniability.
- Consideration of online repudiability during design, analysis and implementation of messaging protocols.
- Active contribution to open-end libraries and adoption by messaging applications.

“My opinions are my own...”

Yay!

- Adversarial examples
- Analysis of existing solutions
- Appeal for usable solutions and not just solutions

Nay!

- Implementation
- Proof summary
- Consistency in definitions.

Questions?

Topics Discussed in this Research Study

- Discussion on Φ_{dre}, closest known solution, its shortcomings and proposed modifications.
- **RSDAKE** - interactive DAKE which improves the performance of Φ_{dre}.
- **SPAWN** - first non-interactive DAKE which provides forward secrecy and also achieves deniability.
- Proposed extension for *TextSecure* Messaging Application.

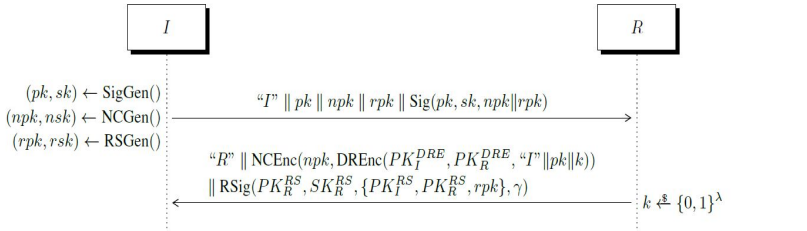


Figure 3: Spawn*. The shared secret is k. γ denotes "R" concatenated with the output of NCEnc. Spawn replaces NCGen with PKGen and NCEnc with PKEnc. In all other ways, Spawn is identical to Spawn*.

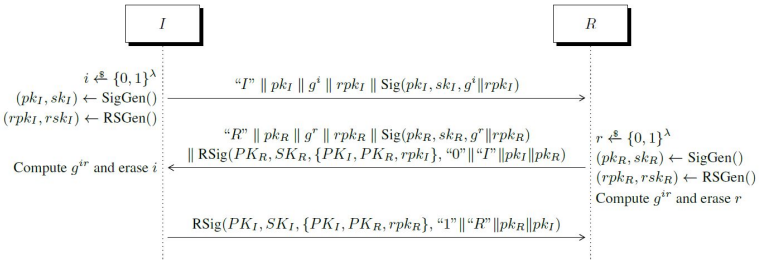


Figure 2: RSDAKE. The shared secret is g^{ir}.

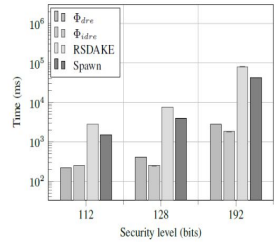


Figure 5: Over a high-bandwidth connection with no latency, the cryptographic overhead of each protocol is clear. The use of ring signatures negatively affects RSDAKE and Spawn.

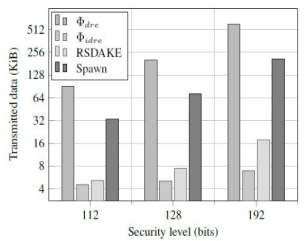


Figure 4: The amount of data transmitted increases significantly with higher security levels. Φ_{dre} and Spawn require significantly more transmissions than Φ_{idre} or RSDAKE.

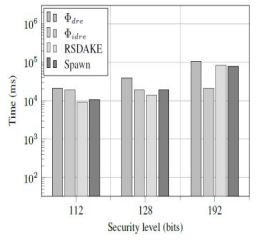


Figure 6: Over a low-bandwidth and high-latency connection, the network significantly affects performance. RSDAKE and Spawn perform the best at 112- and 128-bit security levels.