Partially Blind RSA Signatures

draft-amjad-cfrg-partially-blind-rsa

Ghous Amjad*, Scott Hendrickson, Christopher Wood, Kevin Yeo

IETF 116 - CFRG

Outline

- Motivation
- Background: Blind RSA Signatures
- Partially Blind RSA Signatures
- Benchmarks
- Current Status

Motivation: Blind Signatures

- Privacy Pass
- Web Browsing, e.g.,
  - VPN by Google One
  - iCloud Private Relay
- Avoiding Repeated CAPTCHA Solving
- Private Click Measurement
- Tor DOS Defenses

.

.

Motivation: Partially Blind Signatures

- ‘draft-irtf-cfrg-voprf’ offers partially oblivious variant
- Signatures that can only be used for
  - specific settings
  - specific geographic location etc.
- Avoiding one key per metadata approach
  - May require fixed public metadata choices ahead of time
  - Key management scalability concerns

Motivation: Blind RSA Signatures

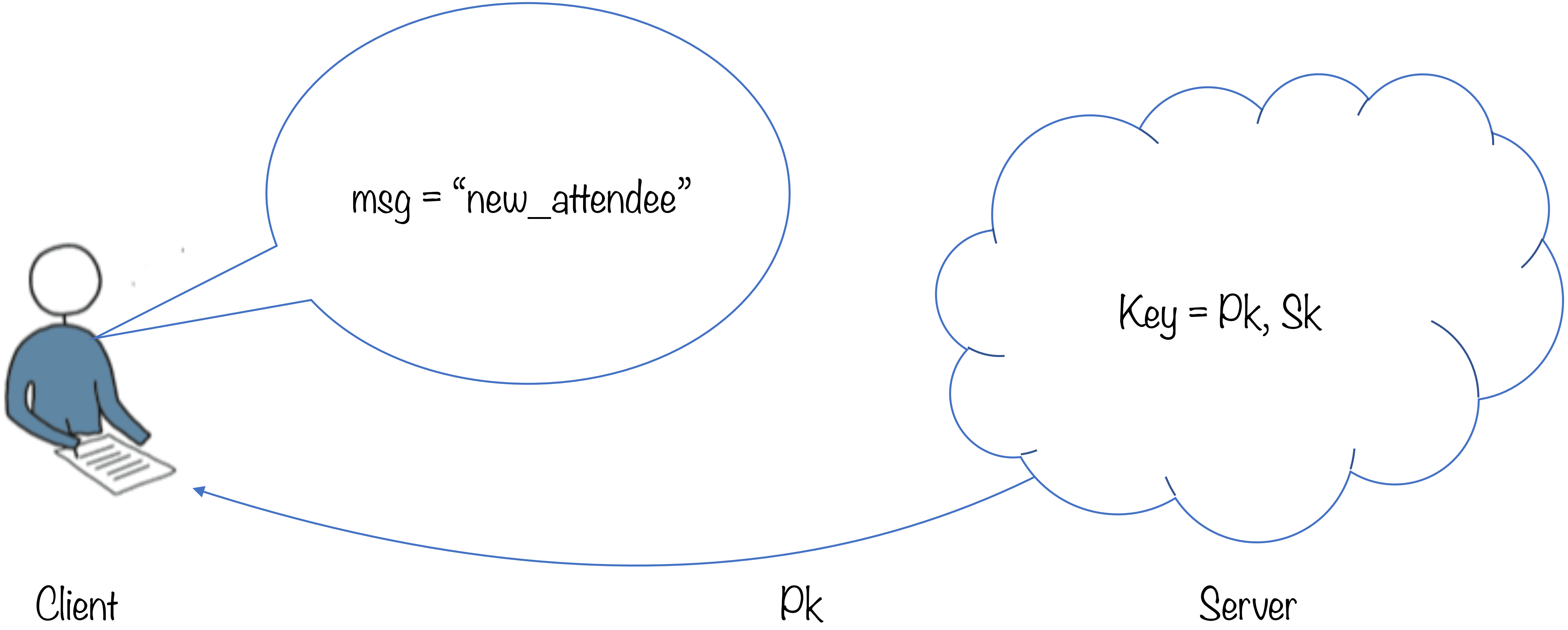
- IETF document adopted for Blind RSA Signatures
  - Simple (one-round scheme, stateless server issuance)
  - Widely supported public verification
  - ‘draft-irtf-cfrg-rsa-blind-signatures’

- Natural to think of Public Metadata support for this standard

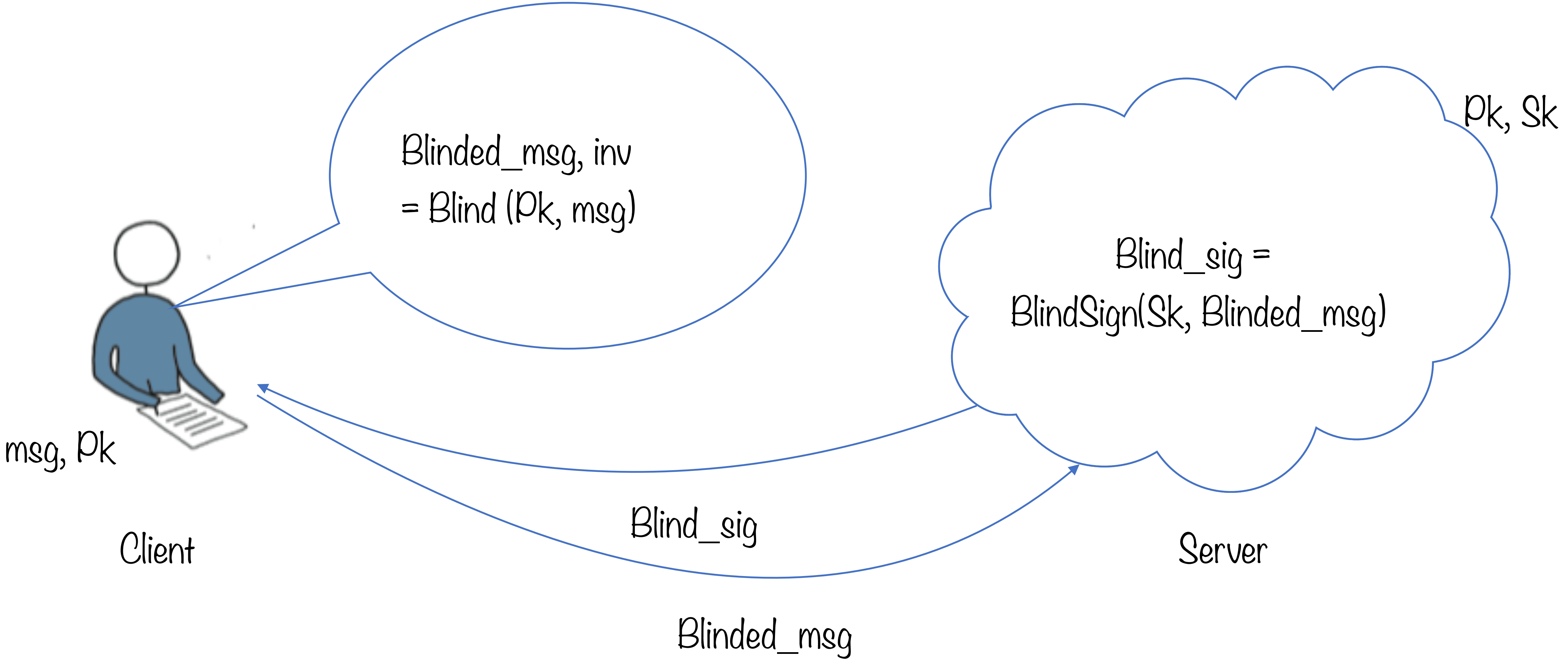
Outline

- Motivation
- Background: Blind RSA Signatures
- Partially Blind RSA Signatures
- Benchmarks
- Current Status

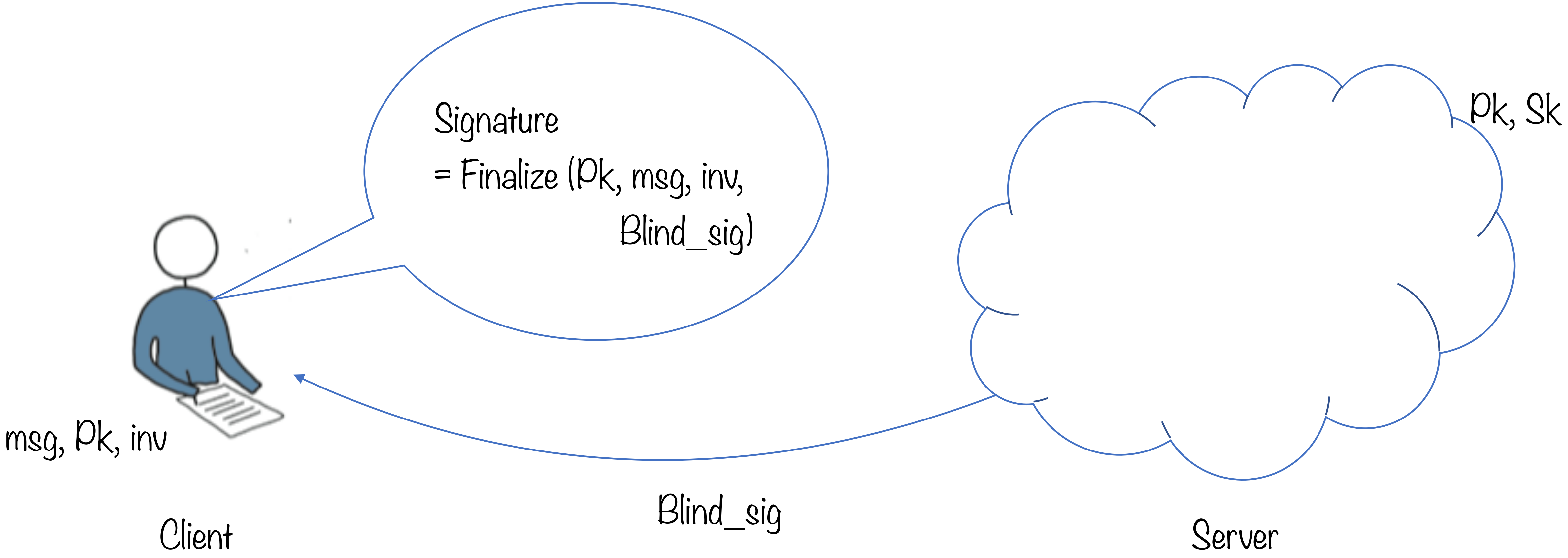
Background: Blind RSA Signatures



Background: Blind RSA Signatures



Background: Blind RSA Signatures



Background: Blind RSA Signatures

- Signature is verified as a sub-routine in Finalize.
- Signature is publicly verifiable.

- Input message is encoded before being blinded.
    - PSS Encoding

Outline

- Motivation
- Background: Blind RSA Signatures
- Partially Blind RSA Signatures
- Benchmarks
- Current Status

Partially Blind RSA Signatures

- Same public metadata (md) needed in all stages of the protocol
  - Blinding
  - Signing
  - Finalizing
  - Verifying

Partially Blind RSA Signatures

- Augmented Input Message
  - Unique encoding of message and “md” passed to PSS encoding
- Augmented Public Key
  - Pk * H(md)
  - using HKDF as H for implementation ease
  - H(md) needs to be co-prime to $\phi(N)$ where N is the RSA modulus, to generate the correct private key
- Generating special RSA modulus
  - N should be a product of two safe primes

Security Considerations

- One-more-unforgeability
- Unlinkability under same public metadata
- Domain separation
  - Different RSA moduli will ensure different augmented public keys for same public metadata
  - Hash functions in input message augmentation and public key augmentation are domain separated
- Denial of Service attacks due to larger public keys

Outline

- Motivation
- Background: Blind RSA Signatures
- Partially Blind RSA Signatures
- Benchmarks
- Current Status

Benchmarks

	Blind RSA Signatures	Partially Blind RSA Signatures*
Blind	459,169 ns	1,695,262 ns
BlindSign	1,298,156 ns	5,368,773 ns
Finalize	37,821 ns	1,262,426 ns

* Timing should improve with more optimized code (e.g. once CRT is used)
* https://github.com/google/anonymous-tokens
* https://github.com/chris-wood/circl/blob/caw/pbrsa/blindsign/blindrsa/pbrsa.go

Outline

- Motivation
- Background: Blind RSA Signatures
- Partially Blind RSA Signatures
- Benchmarks
- Current Status

Current Status

- Two implementations (C++, Go)
- Solves needs in Privacy Pass and related real world applications
- draft-amjad-cfrg-partially-blind-rsa
- Academic paper with security proofs to be put out soon.
- Interest in adopting this document?

Thank you!