Scrappy: SeCure Rate Assuring Protocol with PrivacY

Kosei Akama, Yoshimichi Nakatsuka, Masaaki Sato, Keisuke Uehara

Summarized and reorganized by

Seokwon Oh (swoh@mmlab.snu.ac.kr)

2023-04-18

Network and Distributed System Security (NDSS) Symposium 2024
26 February - 1 March 2024, San Diego, CA, USA
ISBN 1-891562-93-2
https://dx.doi.org/10.14722/ndss.2024.24445
www.ndss-symposium.org

Overview

- Introduction to rate-limiting concept
- Existing major rate-limiting techniques
- Introduction to Scrappy, a new rate-limiting technique
- Design/Implementation
- Evaluation

Introduction to Rate-Limiting [1/3]

https://section.cafe.naver.com/ca-fe/home/create

Introduction to Rate-Limiting [2/3]

.
.
.

보안 절차 *

프로그램을 이용한 자동 개설을 방지하기 위해서
보안절차를 거치고 있습니다.
왼쪽 이미지를 보이는 대로 입력해주세요.

새로고침 음성으로 듣기

정책 동의 * 카페 개인정보보호정책에 동의합니다. 자세히보기

카페를 상거래 목적으로 운영하는 경우, 전자상거래법에 따라 사업자정보를 표시해야 합니다. 자세히보기

취소 만들기

https://section.cafe.naver.com/ca-fe/home/create

Introduction to Rate-Limiting [3/3]

- Online service users access their resources at a moderate rate
  - Malicious users attempt to exceed these limits
  - Online service providers employ techniques to slow down the users
- There are scenarios where rate-limiting plays an important role
  - Online polls and product ratings
  - Services using third-party APIs
  - Services with free trials
  - Preventing dictionary attacks
  - Online crawlers

Existing Major Rate-Limiting Techniques [1/6]

- SMS authentication
  - Authentication via phone numbers
  - Originally not for the rate-limiting purpose

- CAPTCHA
  - Completely automated public Turing test to tell Computers and Humans apart
  - Originally not for the rate-limiting purpose

- Shortcomings
  - Privacy issue
  - Degradation of user experience

Existing Major Rate-Limiting Techniques [2/6]

- CAP (Cryptographic Attestation of Personhood)
  - Users are asked to use their authenticator to sign a challenge
  - The private key is protected by the authenticator's secure element

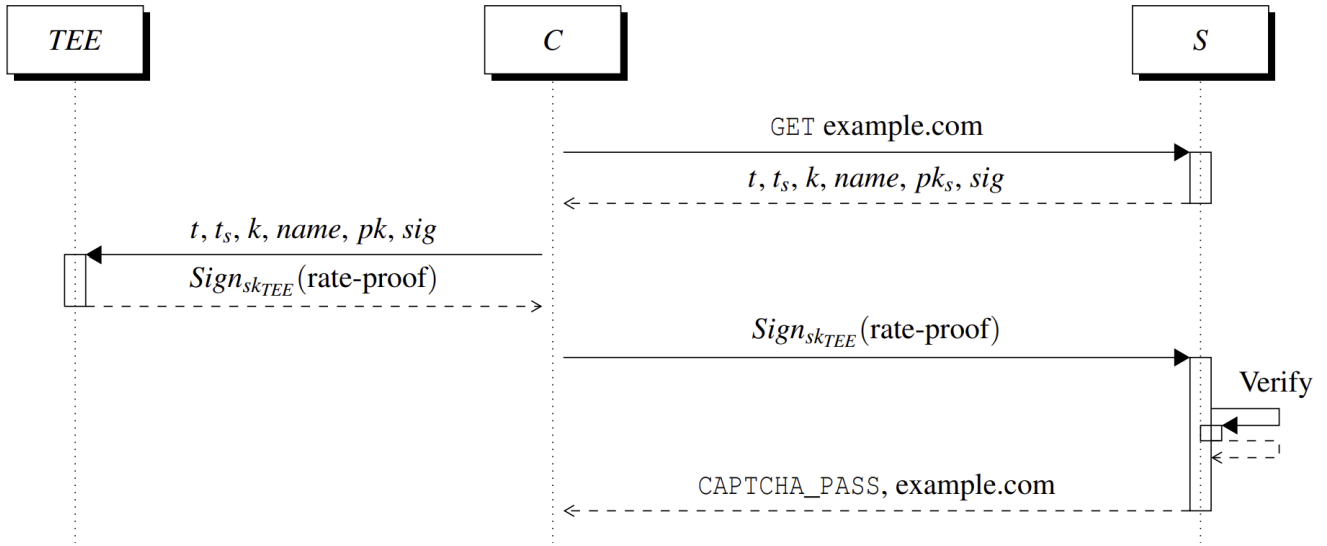
- Shortcomings
  - The security relies on the security of the authenticator's secure element which is a hardware device
  - The secret key is replicated
  - Protecting the secret key within the secure element is crucial

[48] T. Meunier. (March, 2021) Humanity wastes about 500 years per day on CAPTCHAs. It's time to end this madness. Cloudflare Inc. [Online]. Available: https://blog.cloudflare.com/introducing-cryptographic-attestation-of-personhood/

Existing Major Rate-Limiting Techniques [3/6]

- CACTI (CAPTCHA Avoidance via Client-side TEE)
  - It utilizes client-side TEE (Trusted Execution Environment)
  - The TEE provides rate proofs that allow the server to understand the number of user actions conducted in the given time window with a counter

- Shortcoming
  - An adversary is able to forge rate-proofs once the secret key is extracted from the TEE



[51] Y. Nakatsuka, E. Ozturk, A. Paverd, and G. Tsudik, "{CACTI}: Captcha Avoidance via Client-side {TEE} Integration," in 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 2561–2578.

Existing Major Rate-Limiting Techniques [4/6]

- Privacy Pass
  - It allows users to obtain anonymous cryptographic tokens for each task
  - The tokens are implemented using blind signature scheme
  - A user spends each token for each task
- Shortcoming
  - It is vulnerable to time correlation attacks, which use the time difference between the generation and usage of tokens

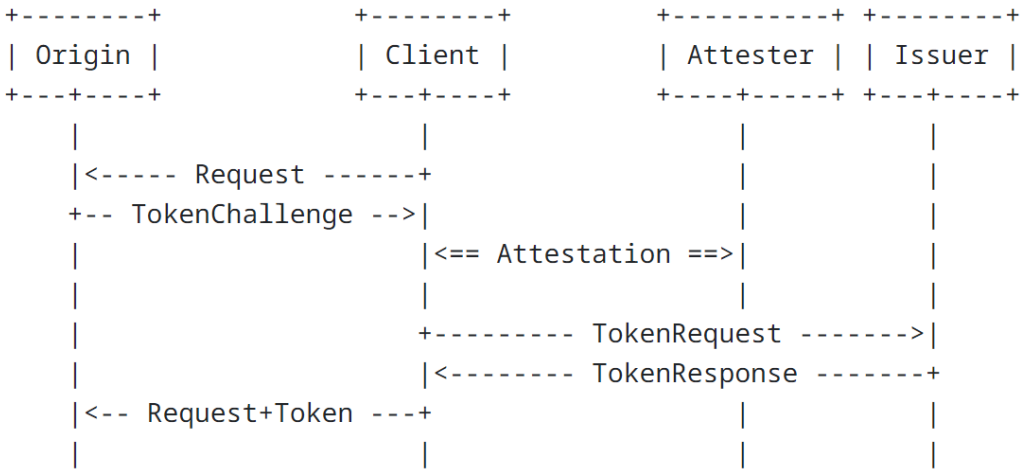


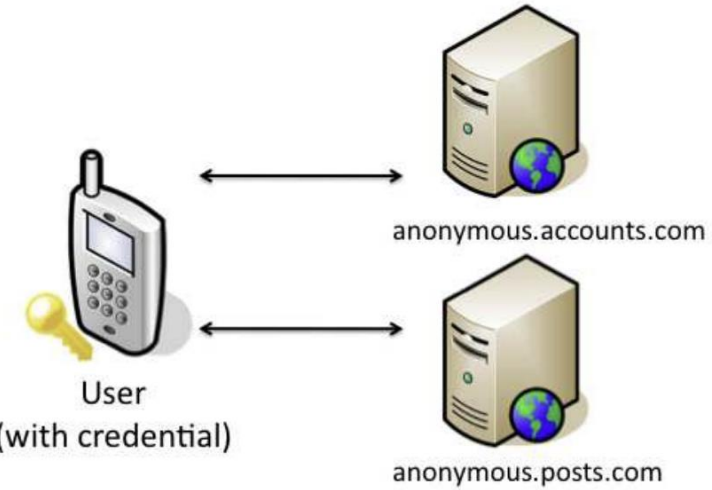
Figure 1: Privacy pass redemption and issuance protocol interaction

[16] C. A. W. Alex Davidson, Jana Iyengar. (2022, October) draft-ietfprivacypass-architecture-08 - The Privacy Pass Architecture. Internet Engineering Task Force. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-privacypass-architecture/

Existing Major Rate-Limiting Techniques [5/6]

- Opaak (OPen Anonymous Authentication frameworK)
  - It provides rate-limiting for mobile phone users
  - A user obtains an anonymous credential upon presenting the phone number

- Shortcomings
  - It assumes that a user cannot possess many phone numbers
  - It encrypts the private key using a user-defined password
  - The private key gets exposed to untrusted memory



[47] G. Maganis, E. Shi, H. Chen, and D. Song, "Opaak: using mobile phones to limit anonymous identities online," in Proceedings of the 10th international conference on Mobile systems, applications, and services, 2012, pp. 295–308.

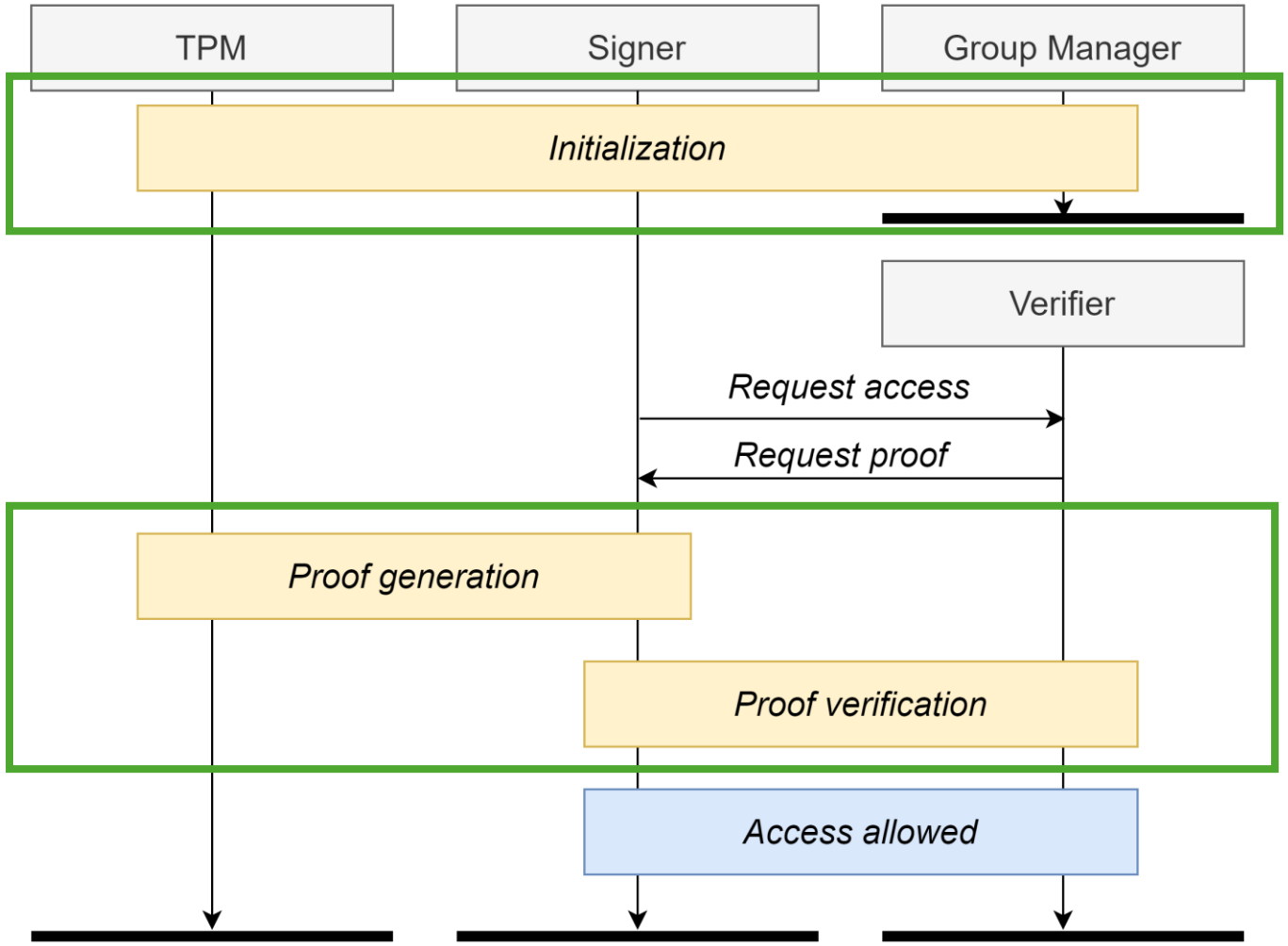
Existing Major Rate-Limiting Techniques [6/6]

	Privacy issue	User experience	Private key storage	Timing-corr attack	HW dependency
CAPTCHA		X			
SMS	X	X			
CAP					X
CACTI					X
Privacy Pass				X	
Opaak			X		
Scrappy					

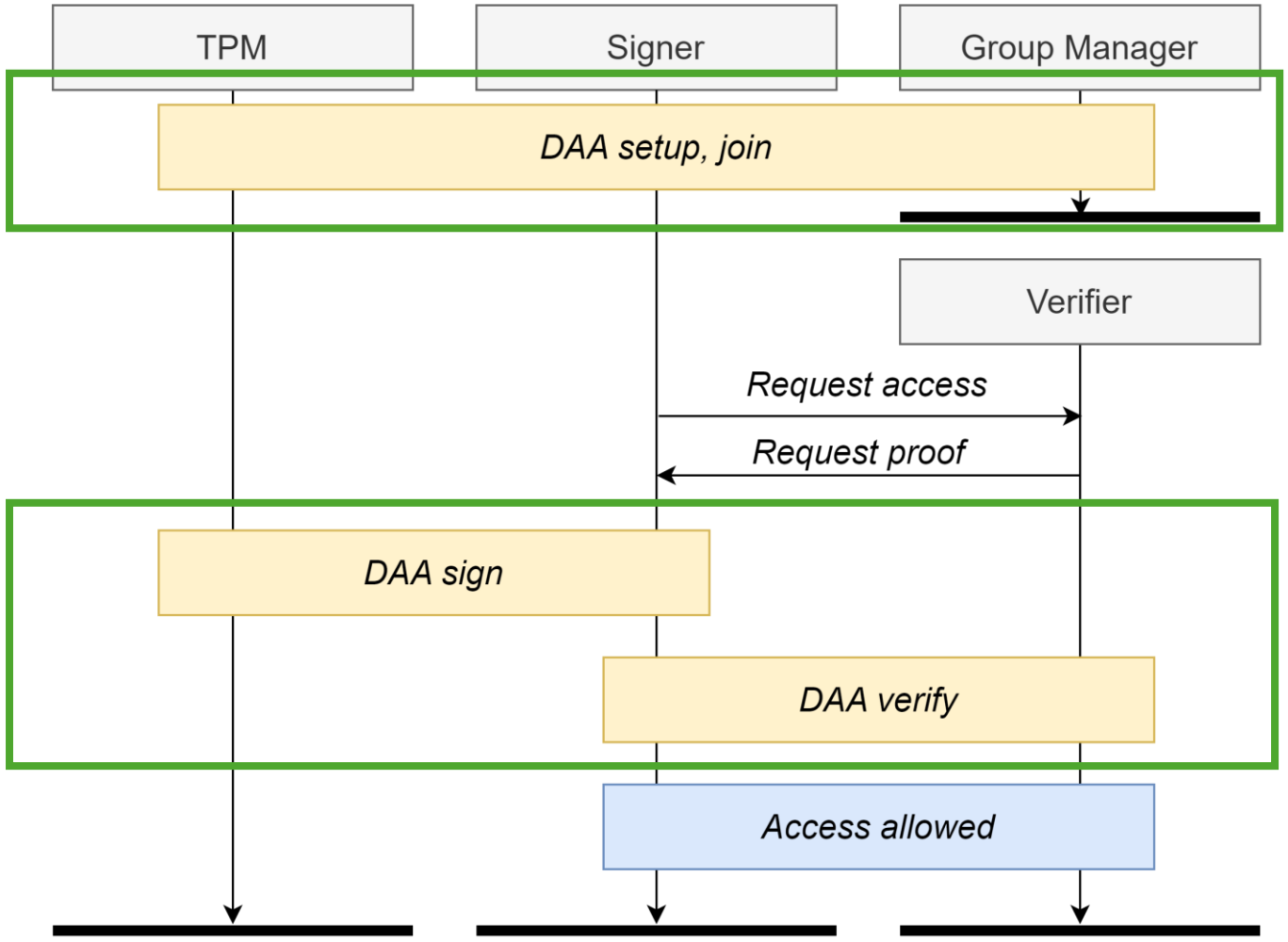
Scrappy

- Scrappy
  - A new rate-limiting protocol this paper proposes
  - An acronym of SeCure Rate Assuring Protocol with PrivacY
- Security requirements of Scrappy
  - Rate-limiting
  - Unforgeability
  - Unlinkability
- Base techniques under Scrappy
  - Group signature scheme
  - Direct Anonymous Attestation (DAA)
  - Trusted Platform Module (TPM)

Scrappy Protocol Overview [1/2]

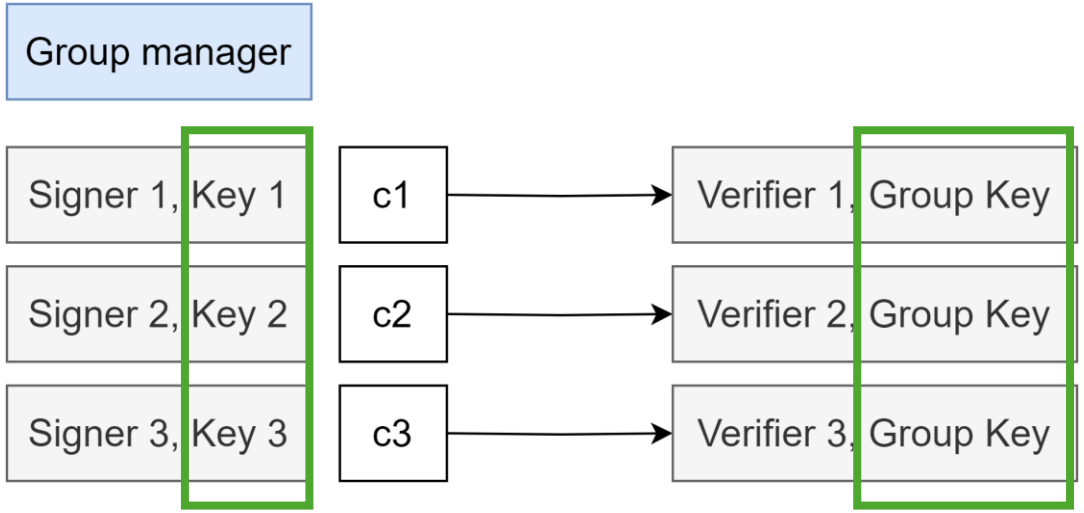


Scrappy Protocol Overview [2/2]



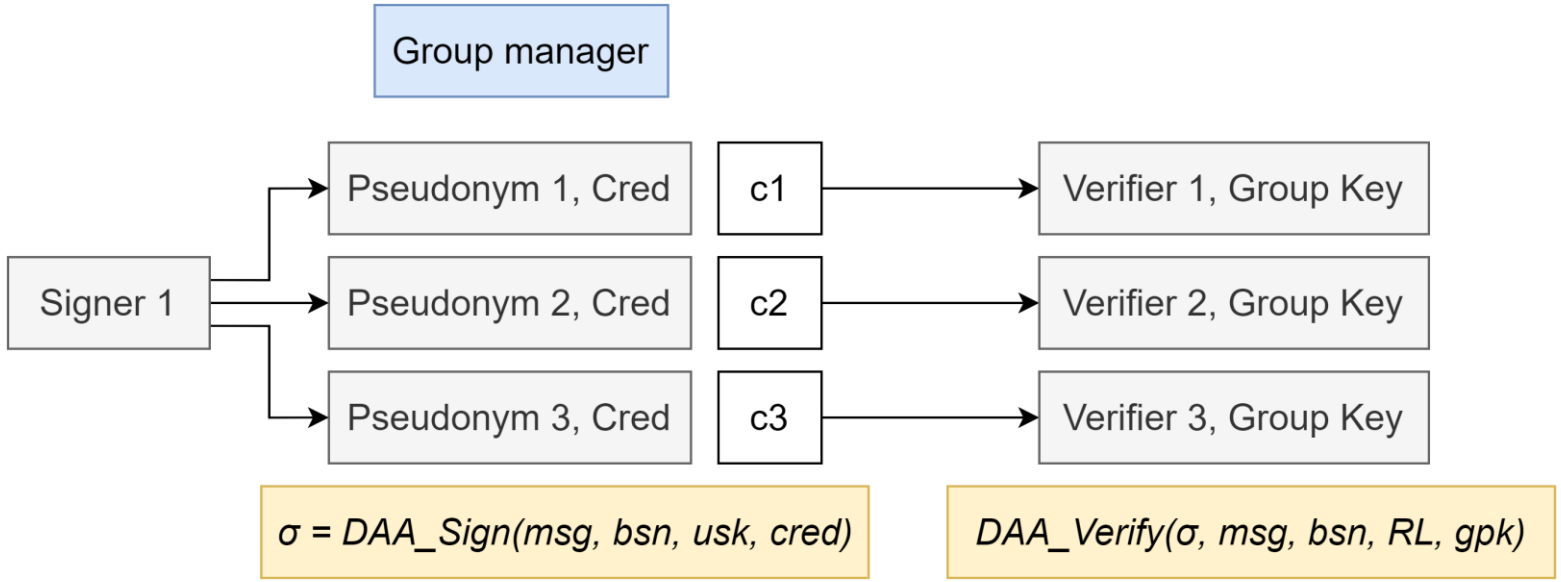
Group Signature Scheme

- Group signature allows signers to prove group membership
  - There are multiple private keys with a single public key
  - Signers cannot be distinguished by their signatures
  - It consists of group manager, signer, verifier



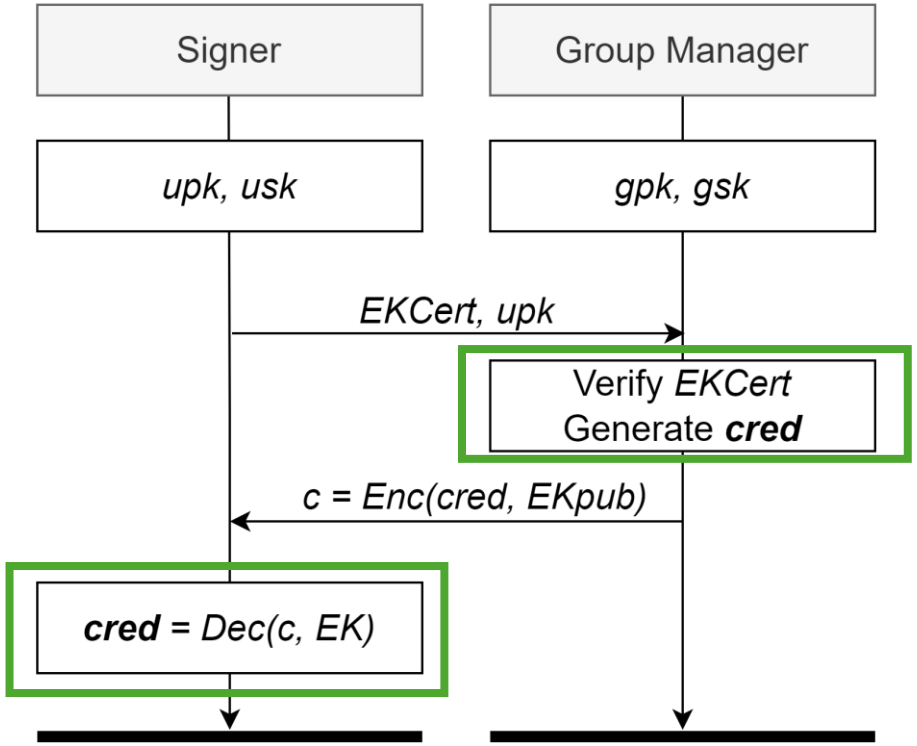
Direct Anonymous Attestation

- Direct Anonymous Attestation (DAA) is a group signature scheme with additional properties
  - It provides anonymity, unlinkability, and pseudonymity
  - It is adopted by Trusted Computing Group in the TPM specification

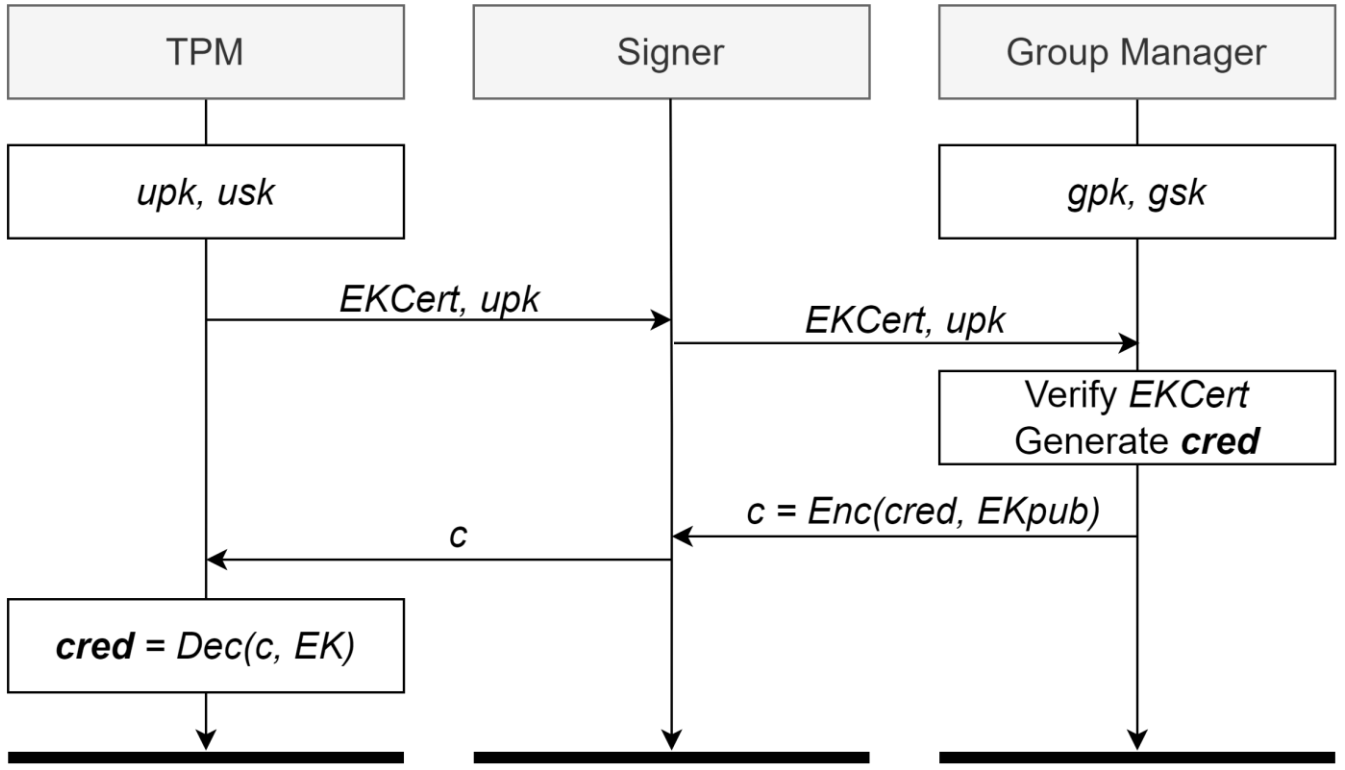


[19] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," in Proceedings of the 11th ACM conference on Computer and communications security, 2004, pp. 132–145.

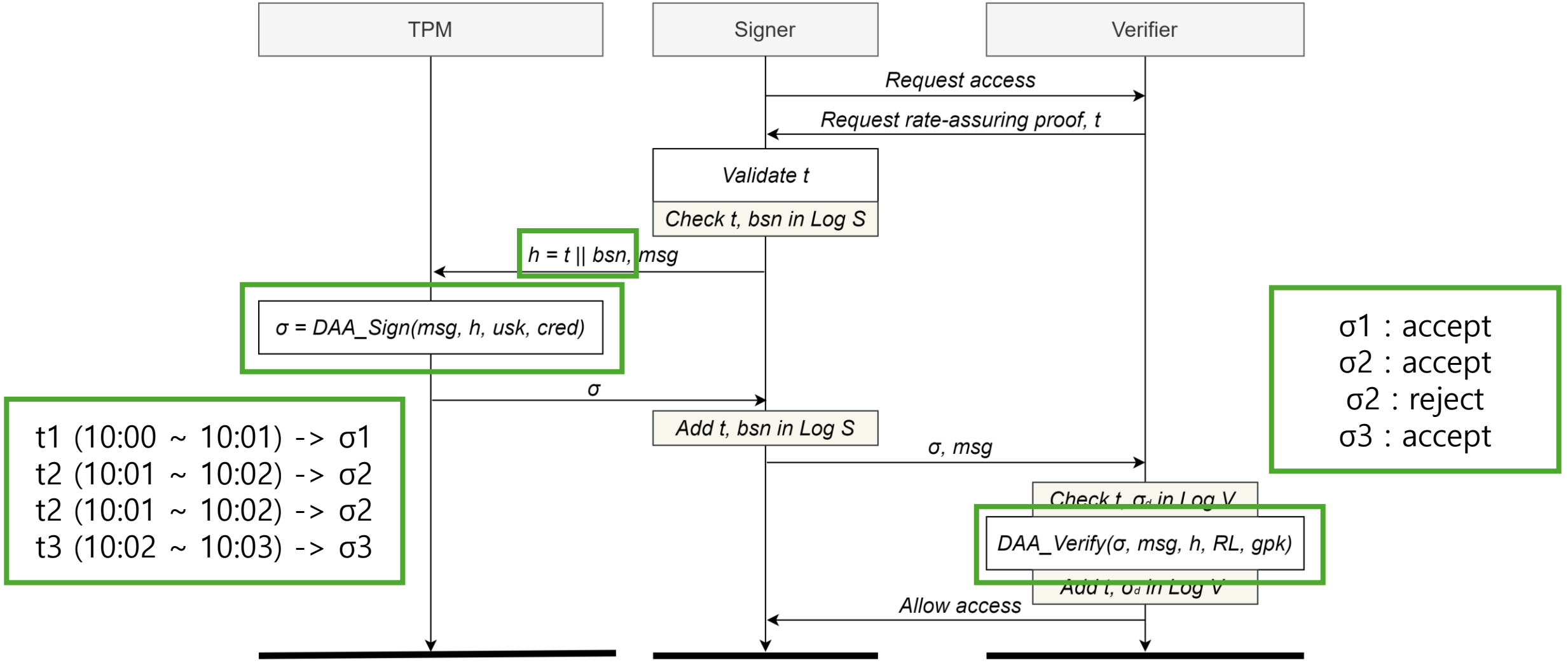
Direct Anonymous Attestation : Setup, Join



Scrappy Protocol : Initialization



Scrappy Protocol : Rate-Limiting



Implementation

- Browser extension
  - It receives/sends the data between the verifier server, and signer application
- Signer application
  - It is implemented in Golang (high performance, memory safety)
  - It uses SQLite
  - It uses a modified version of the go-tpm library
- Verifier server
  - It is implemented in Golang
  - It uses SQLite
  - It is set to allow one access per minute
  - It sends the data necessary for the rate-assuring proof

Performance Evaluation

- Latency evaluation
  - It is reasonable as Scrappy uses a resource-limited TPM
  - CAP is excluded as it requires physical action
  - Opaak is excluded as its latency drastically varies across environments

Work	Proof generation [ms]	Proof verification [ms]
Scrappy	243.16	84.1
CACTI	211.9	27.3
Privacy Pass (N tokens)	341.48 + 180.87 * N	57.8

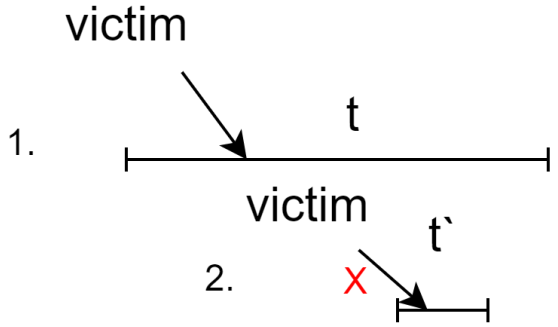
- Storage evaluation
  - 100,000 entries in the verifier side log : 6.64 MB
  - It stores a small amount for a large log

Security Evaluation [1/3]

Rate-limiting	Unforgeability	Unlinkability
Generating multiple proofs upfront	**Signature forgery attacks**	Signer tracking via proofs
Obtaining multiple credentials	Timestamp forgery attacks	**Signer tracking via t**
Network attacks	Device reset attacks	The use of basenames in Scrappy and its privacy implications
Compromised devices		Violating signer privacy via side-channel attacks
		Rogue GM
		GM with only one member

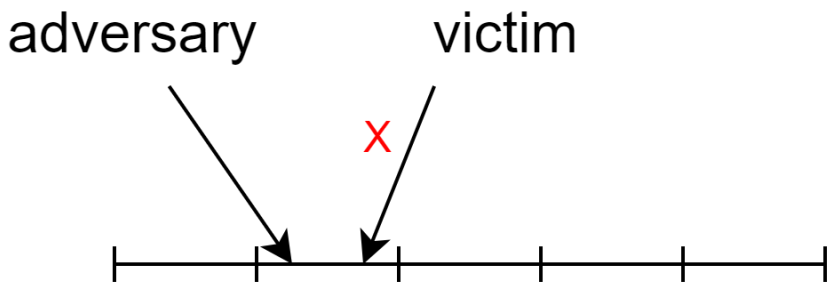
Security Evaluation [2/3]

- Signature forgery attacks
  - A malicious signer may attempt to generate a fake rate-assuring proof
  - It is impossible for signers to receive a valid cred from GM without TPM
- Signer tracking via t
  - A malicious verifier gives a long t
  - The malicious verifier gives t` that overlaps with t
  - It is detectable since the signer observes overlapping time windows



Security Evaluation [3/3]

- Compromised devices
  - Malicious signers may attempt to extract the EK, usk from TPM
  - Unforgeability, and unlinkability properties are violated
  - The verifier will accept a proof generated by either adversary or victim



Usability Analysis

- Installation
  - It requires browser extension, and signer application to be downloaded
  - It can be improved with integrating the logic directly into the browser
- Signer-perceived latency
  - It is comparable to other rate-limiting systems
  - It does not degrade user experience
- Signer involvement
  - It is not required
  - Still, it improves security
  - We can give the signers the choice

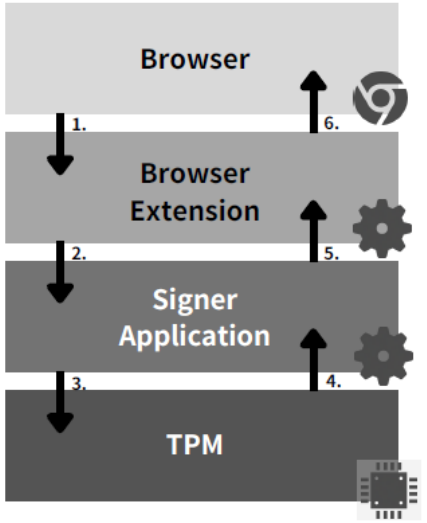
Conclusion

- There are various techniques to ensure rate-limiting
- The paper proposes Scrappy, a new rate-limiting protocol
- Scrappy is unforgeable, and privacy-preserving
- Scrappy utilizes DAA protocol, supported by TPM
- Scrappy does not rely on the security of the hardware device

Appendix 1 : TPM (Trusted Platform Module)

- TPM is an embedded computer chip that can securely store artifacts
  - It includes passwords, certificates, or encryption keys
  - The embedded endorsement key proves the uniqueness of the platform
  - It also provides security-related functions such as *TMP2_Sign*
  - It was conceived by Trusted Computing Group

- TPM is the first choice for Scrappy
  - It supports DAA by default
  - Microsoft officially announced that the Window 11 OS requires it



Appendix 2 : Other Implementations