Short Double- and N-times-Authentication-Preventing Signatures from ECDSA and More

David Derler[‡], **Sebastian Ramacher**[‡], Daniel Slamanig[§]

EuroS&P'18, April 25, 2018

‡ TU Graz § AIT Austrian Institute of Technology

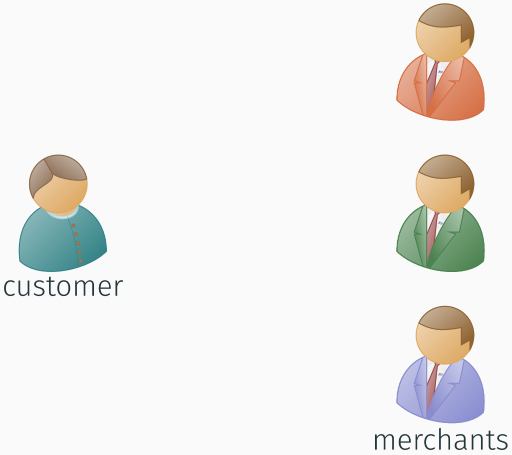
Motivation

Digital Signatures

Applications

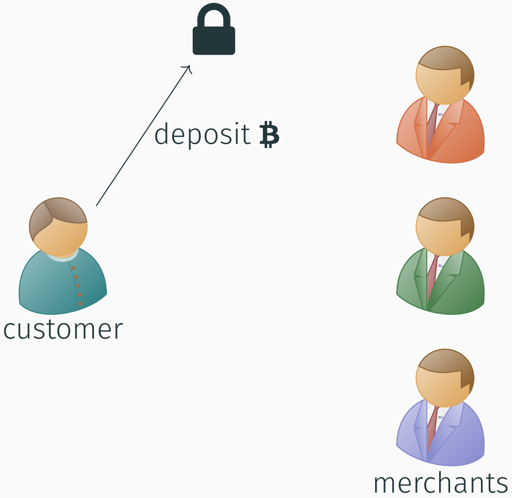
- Signing transactions in cryptocurrencies
- Certificate and software signing
- And many more

Penalize Double-Spending

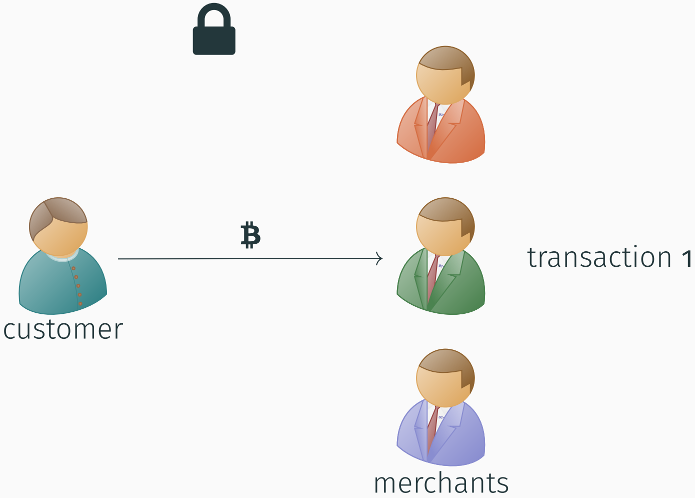


Penalize Double-Spending

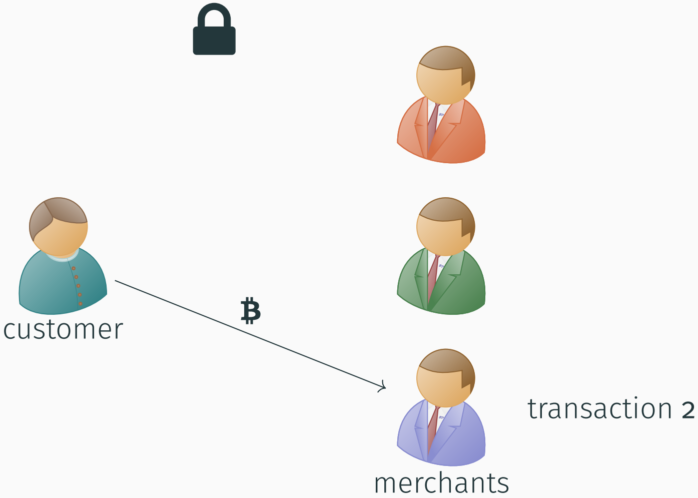
create offline payment channel



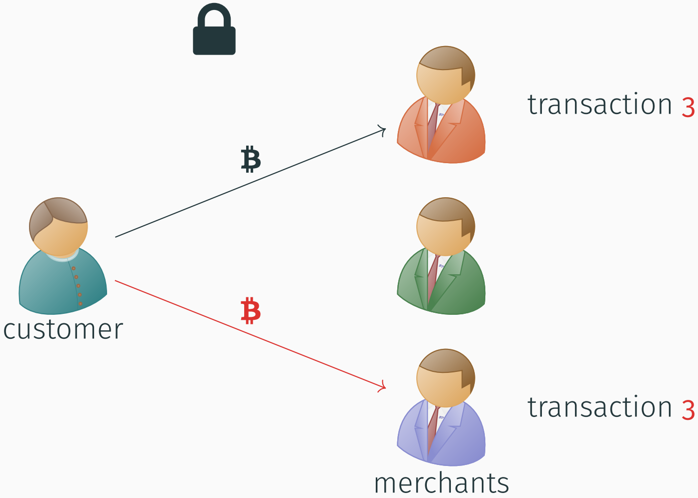
Penalize Double-Spending



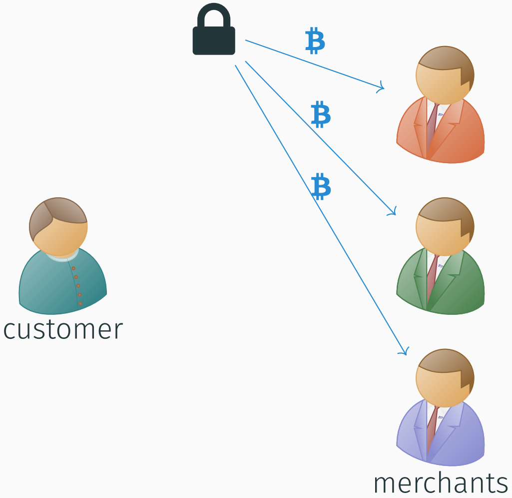
Penalize Double-Spending



Penalize Double-Spending

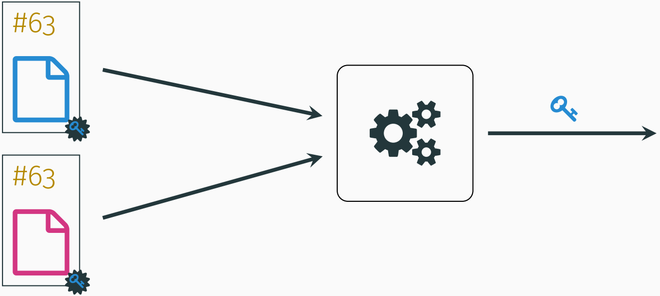


Penalize Double-Spending



receive deposit
on misuse

Double-Authentication Preventing Signatures [PS14]



- Same context, different content

» Can extract secret key

- Extraction from honest and malicious keys

Existing DAPS

Existing schemes

- Factoring based [PS14, PS17, BPS17]
- DLOG based [RKS15]
- All of them based on trapdoor properties

Existing DAPS

Existing schemes

- Factoring based [PS14, PS17, BPS17]
- DLOG based [RKS15]
- All of them based on trapdoor properties

Problems:

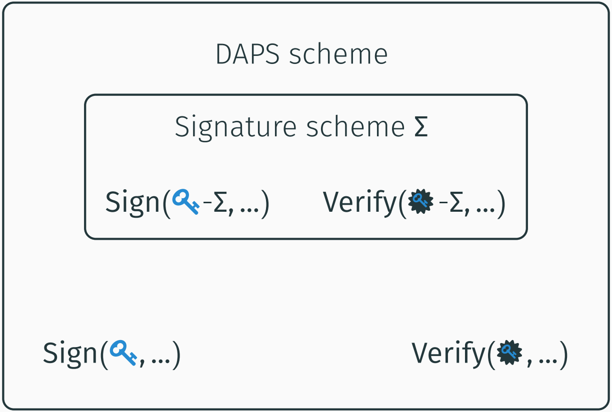
- Factoring based: not compatible with plain RSA signatures
- DLOG based: inefficient

Can we build efficient DAPS from existing signature schemes in a black-box way?

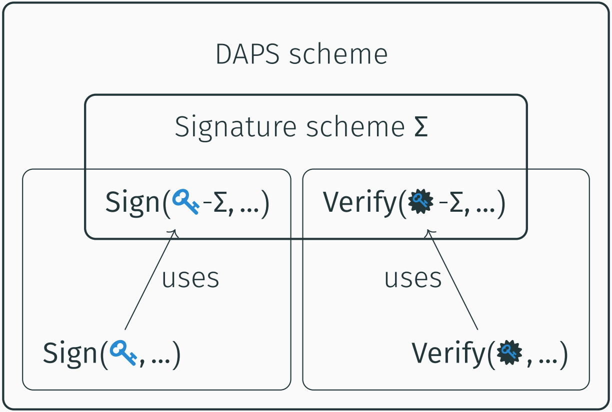
Black-box Extension

Signature scheme Σ

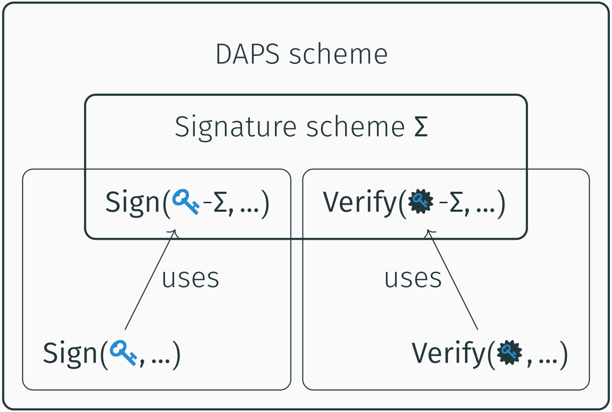
Sign(-Σ, …) Verify(-Σ, …)


DAPS scheme
Signature scheme Σ
Sign(-Σ, ...)
Verify(-Σ, ...)
Sign(, ...)
Verify(, ...)

Black-box Extension



Black-box Extension



DAPS secret key contains Σ secret key

Observations

Extraction of Σ secret key often sufficient

- ✔ Example: ECDSA key protecting Bitcoin deposit

Observations

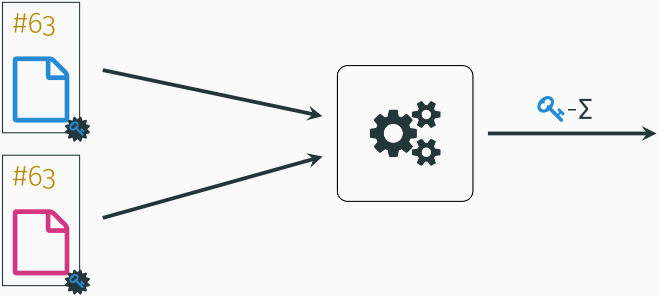
Extraction of Σ secret key often sufficient

- ✔ Example: ECDSA key protecting Bitcoin deposit
- » New security notions covering Σ secret key extraction
- + for honest and malicious keys

Observations

Extraction of Σ secret key often sufficient

- ✔ Example: ECDSA key protecting Bitcoin deposit
- » New security notions covering Σ secret key extraction
- + for honest and malicious keys



Observations

Extraction of Σ secret key often sufficient

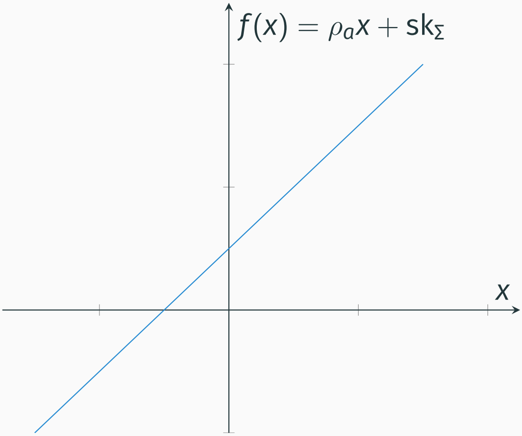
- ✔ Example: ECDSA key protecting Bitcoin deposit
- » New security notions covering Σ secret key extraction
- + for honest and malicious keys

Most applications

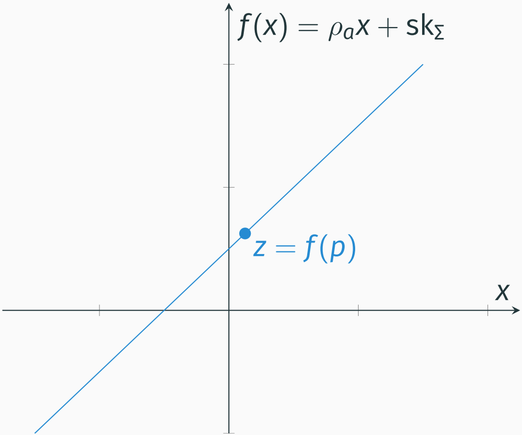
- Polynomial address space sufficient

Construction

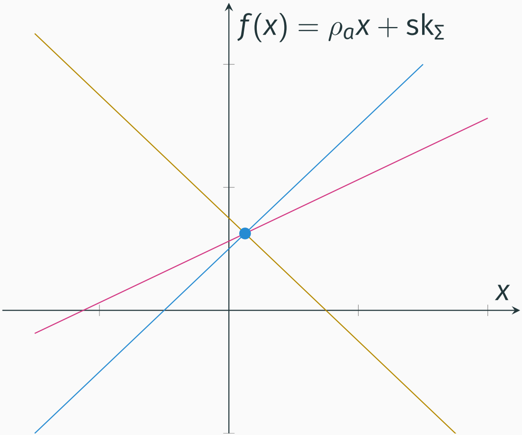
Shamir Secret Sharing



Shamir Secret Sharing

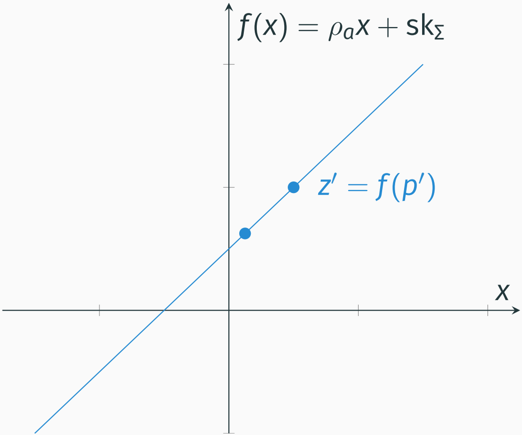


Shamir Secret Sharing



- One point reveals nothing about sk_Σ

Shamir Secret Sharing



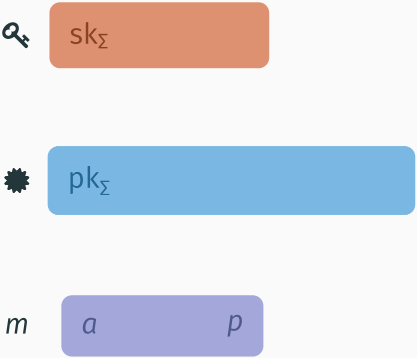
- One point reveals nothing about sk_Σ
- Two points allow to recover sk_Σ

Generic DAPS

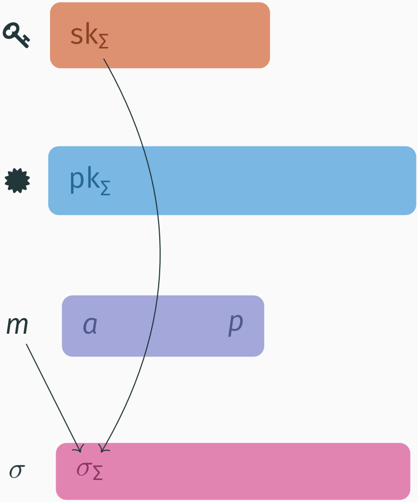
sk_Σ

pk_Σ

Generic DAPS

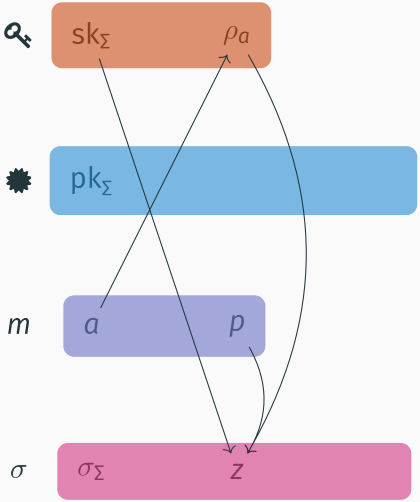


Generic DAPS



signature on m

Generic DAPS



secret sharing of sk_Σ

Generic DAPS

$sk_\Sigma \quad \rho_a$

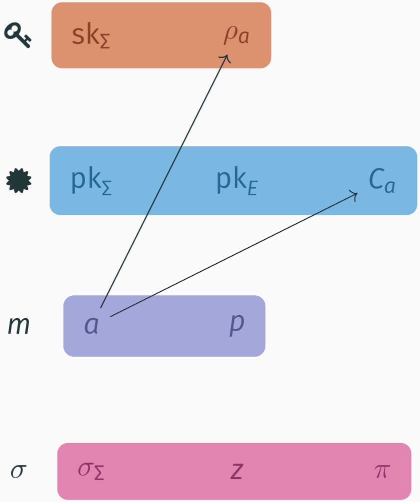
$pk_\Sigma \quad pk_E \longrightarrow C_a$

encrypt coefficients

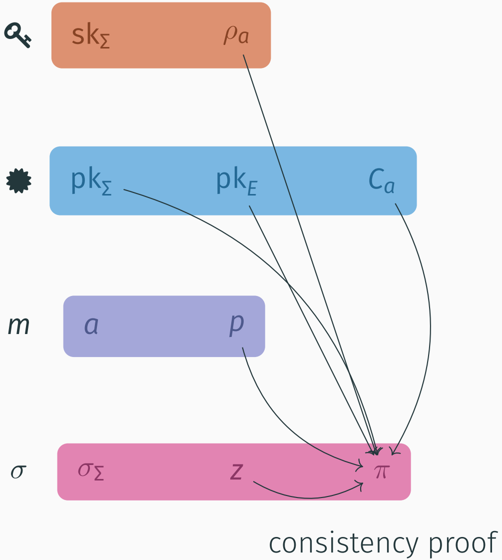
m: $a \quad p$

σ: $\sigma_\Sigma \quad z$

Generic DAPS



Generic DAPS



Generic DAPS: Wrap Up

Generic approach:

- Black-box use of Σ
- **+** Verifiable Shamir secret sharing of Σ secret key
- **+** Sharing polynomial determined by address

$$f(x) = \rho_a x + \mathsf{sk}_\Sigma$$

Generic DAPS: Wrap Up

Generic approach:

- Black-box use of Σ
- \+ Verifiable Shamir secret sharing of Σ secret key
- \+ Sharing polynomial determined by address

$$f(x) = \rho_a x + \mathsf{sk}_\Sigma$$

- Evaluate verification relation in encrypted domain
- Zero-knowledge proof of consistency

Generic DAPS: Wrap Up

Generic approach:

- Black-box use of Σ
- + Verifiable Shamir secret sharing of Σ secret key
- + Sharing polynomial determined by address

$$f(x) = \rho_a x + \mathsf{sk}_\Sigma$$

- Evaluate verification relation in encrypted domain
- Zero-knowledge proof of consistency

Can prove unforgeability via unforgeability of Σ (black-box)

Generic DAPS: Wrap Up (cont)

- For example, applies to ECDSA, EdDSA, DSA
- Short DAPS signatures
- Public key linear in size of address space
  (contains encrypted sharing polynomials per address)

Generic DAPS: Wrap Up (cont)

- For example, applies to ECDSA, EdDSA, DSA
- Short DAPS signatures
- Public key linear in size of address space
  (contains encrypted sharing polynomials per address)
- Extendable to N-authentication preventing signatures
  - » Use degree $N - 1$ sharing polynomial

Implementation

- Easy extension of existing implementations
- + Implement secret sharing
- + Implement consistency proof
- ✔ We provide implementation in OpenSSL

Implementation

- Easy extension of existing implementations
- + Implement secret sharing
- + Implement consistency proof
- ✔ We provide implementation in OpenSSL

Scheme	Sign [ms]	Verify [ms]	$\lvert sk \rvert$ [bits]	$\lvert pk \rvert$ [bits]	$\lvert \sigma \rvert$ [bits]
ECDSA-DAPS $(\mathtt{s})$	0.76	1.33	$256 \cdot (1 + 2n)$	$514 \cdot (1 + n)$	1280
ECDSA-DAPS $(\mathtt{p})$	0.23	0.35	$256 \cdot (1 + 2n)$	$514 \cdot (1 + n)$	1280
ECDSA $(\mathtt{s})$	0.09	0.35	256	257	512
ECDSA $(\mathtt{p})$	0.06	0.21	256	257	512

Table 1: Runtime and sizes; `secp256k1` $(\mathtt{s})$, `prime256v1` $(\mathtt{p})$

Conclusion

Conclusion

Contribution

- ✔ Generic construction
- ✔ Can extend virtually all DLOG-based signature schemes
- ✔ Focus on extraction of underlying signature scheme key
- ✔ Shortest black-box DAPS
  (slightly weaker, yet very reasonable model)
- ✔ Extendable to N-authentication preventing signatures

Conclusion

Contribution

- ✔ Generic construction
- ✔ Can extend virtually all DLOG-based signature schemes
- ✔ Focus on extraction of underlying signature scheme key
- ✔ Shortest black-box DAPS

  (slightly weaker, yet very reasonable model)
- ✔ Extendable to N-authentication preventing signatures

Follow-up work [Poe18]

- Even shorter DAPS (non-black-box)

Conclusion

Contribution

- ✔ Generic construction
- ✔ Can extend virtually all DLOG-based signature schemes
- ✔ Focus on extraction of underlying signature scheme key
- ✔ Shortest black-box DAPS
  (slightly weaker, yet very reasonable model)
- ✔ Extendable to N-authentication preventing signatures

Follow-up work [Poe18]

- Even shorter DAPS (non-black-box)

Future work

- Reduce public key overhead per address

Questions?

Implementation: `https://github.com/IAIK/daps-dl`

Supported by: prisma cloud

References i

[BPS17] Mihir Bellare, Bertram Poettering, and Douglas Stebila. Deterring certificate subversion: Efficient double-authentication-preventing signatures. In *PKC*, 2017.

[Poe18] Bertram Poettering. Shorter double-authentication preventing signatures for small address spaces. In *AFRICACRYPT*, volume 10831 of *Lecture Notes in Computer Science*, pages 344–361. Springer, 2018.

[PS14] Bertram Poettering and Douglas Stebila. Double-authentication-preventing signatures. In *ESORICS*, 2014.

[PS17] Bertram Poettering and Douglas Stebila. Double-authentication-preventing signatures. *Int. J. Inf. Sec.*, 16(1), 2017.

[RKS15] Tim Ruffing, Aniket Kate, and Dominique Schröder. Liar, liar, coins on fire!: Penalizing equivocation by loss of bitcoins. In *ACM CCS*, 2015.