A group signature scheme in the generic group model

Rémi Clarisse (joint work with Olivier Sanders)

CARAMBA seminar – January 13th, 2020

IRMAR – Univ. de Rennes
Orange Labs – Cesson-Sévigné

Table of contents

Building blocks

Public Key Encryption

Public key encryption scheme

- $\Gamma.\texttt{Keygen}(1^\lambda) \to [\mathsf{sk}, \mathsf{pk}]$
- $\Gamma.\texttt{Encrypt}(\mathsf{pk}, m) \to c$
- $\Gamma.\texttt{Decrypt}(\mathsf{sk}, c) \to \{m, \perp\}$

IND-CCA2 security

indistinguishability under adaptive chosen ciphertext attacks

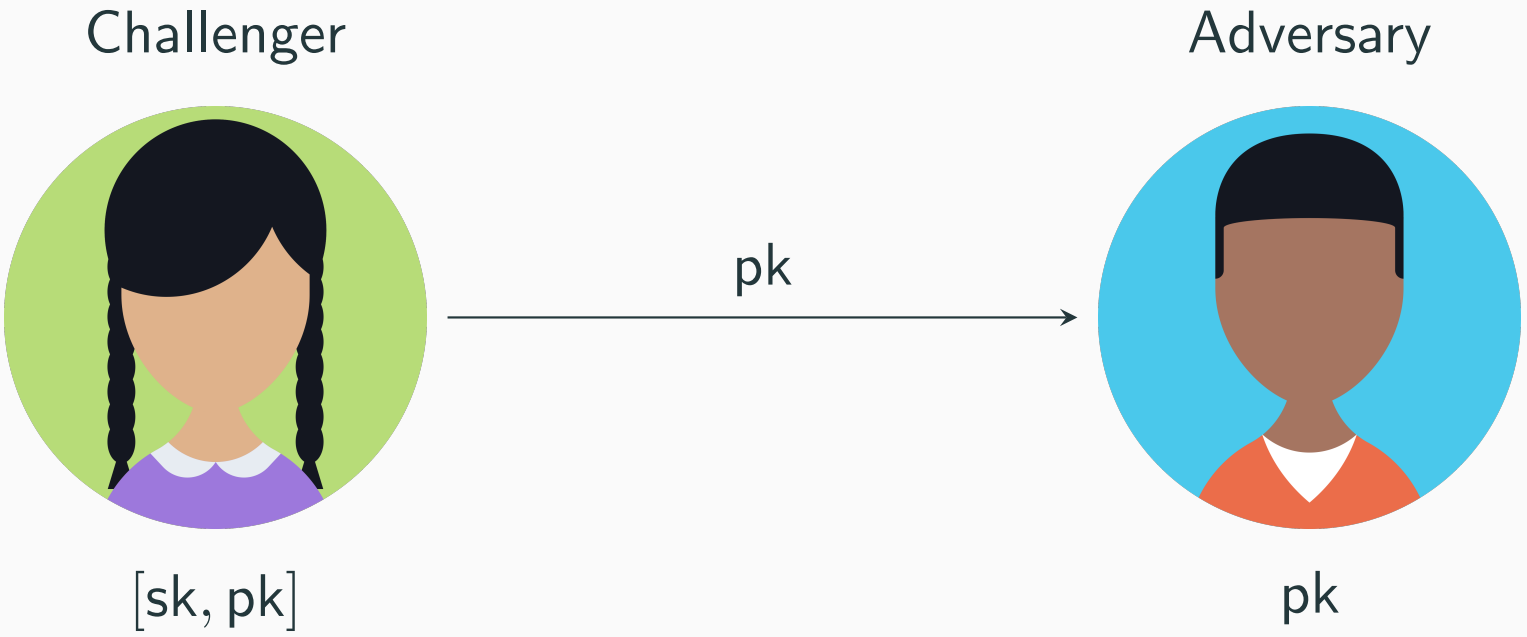
IND-CCA2

Challenger

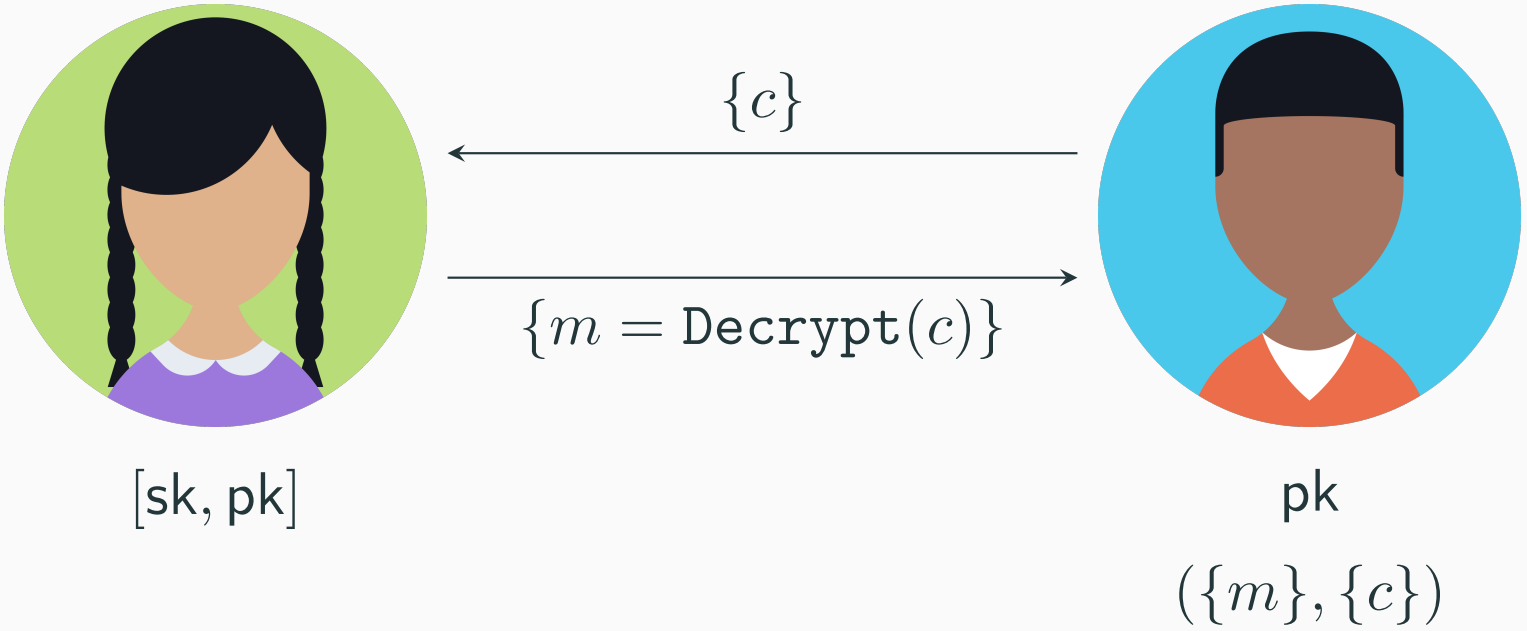
$[sk, pk]$

Adversary

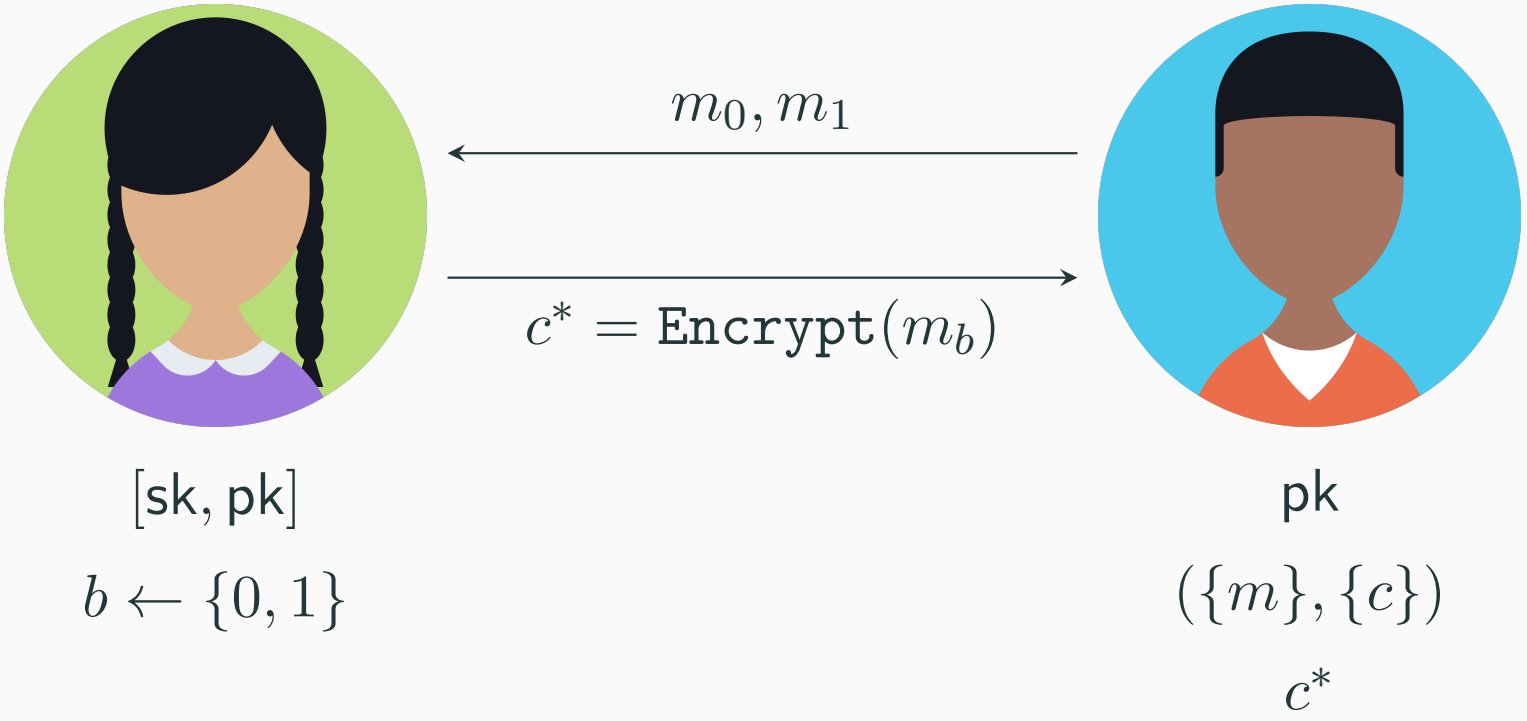
IND-CCA2



IND-CCA2



IND-CCA2


Challenger
Adversary
m_0, m_1
$c^* = \texttt{Encrypt}(m_b)$
[sk, pk]
$b \leftarrow \{0, 1\}$
pk
$(\{m\}, \{c\})$
c^*

IND-CCA2

Challenger

Adversary

$\{c' \neq c^*\}$

$\{m' = \texttt{Decrypt}(c')\}$

$[\mathsf{sk}, \mathsf{pk}]$

$b \leftarrow \{0, 1\}$

pk

$(\{m\}, \{c\})$

c^*

$(\{m'\}, \{c'\})$

IND-CCA2

Challenger

Adversary

b^*

is $b^* = b$?

[sk, pk]

$b \leftarrow \{0, 1\}$

pk

$(\{m\}, \{c\})$

c^*

$(\{m'\}, \{c'\})$

Digital Signature

Digital signature scheme

- $\Sigma.\mathtt{Setup}(1^\lambda) \to pp$
- $\Sigma.\mathtt{Keygen}(pp) \to [\mathsf{sk}, \mathsf{pk}]$
- $\Sigma.\mathtt{Sign}(\mathsf{sk}, m) \to \sigma$
- $\Sigma.\mathtt{Verify}(\mathsf{pk}, m, \sigma) \to \{0, 1\}$

EUF-CMA security [GMR88]

existential unforgeability under chosen message attacks

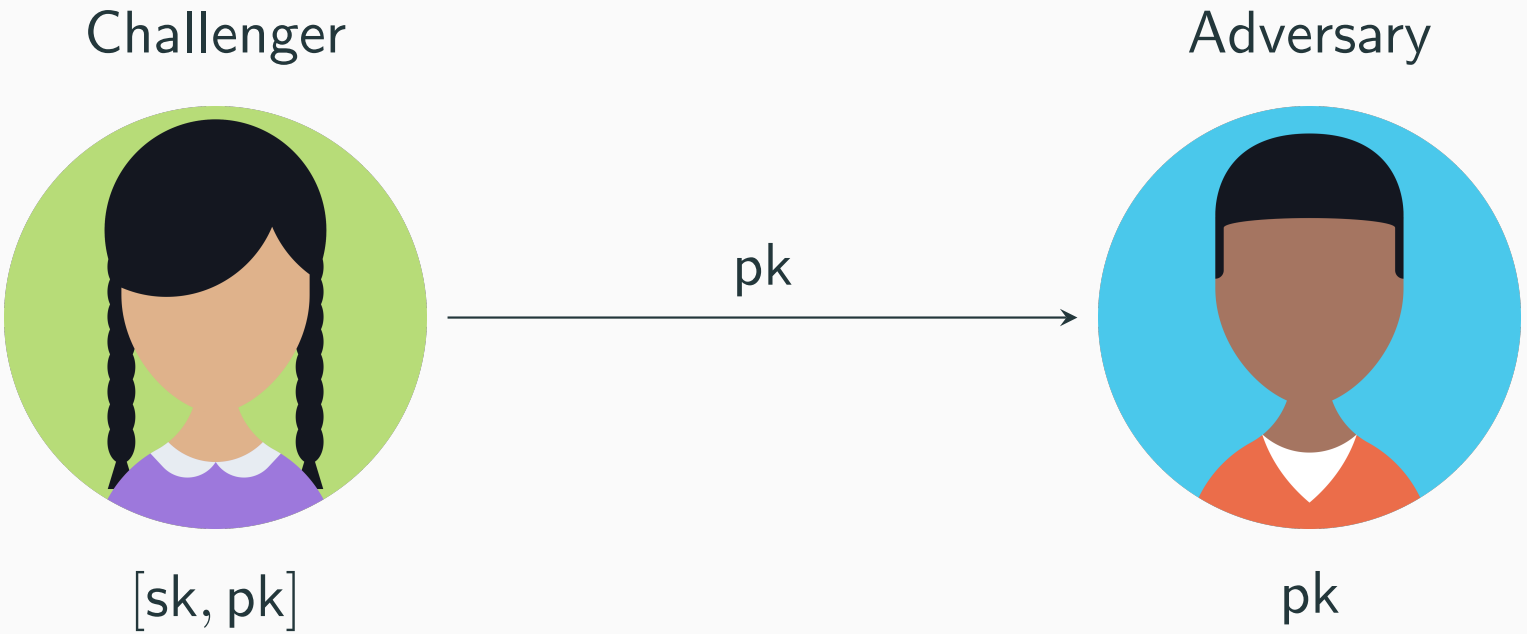
EUF-CMA [GMR88]

Challenger

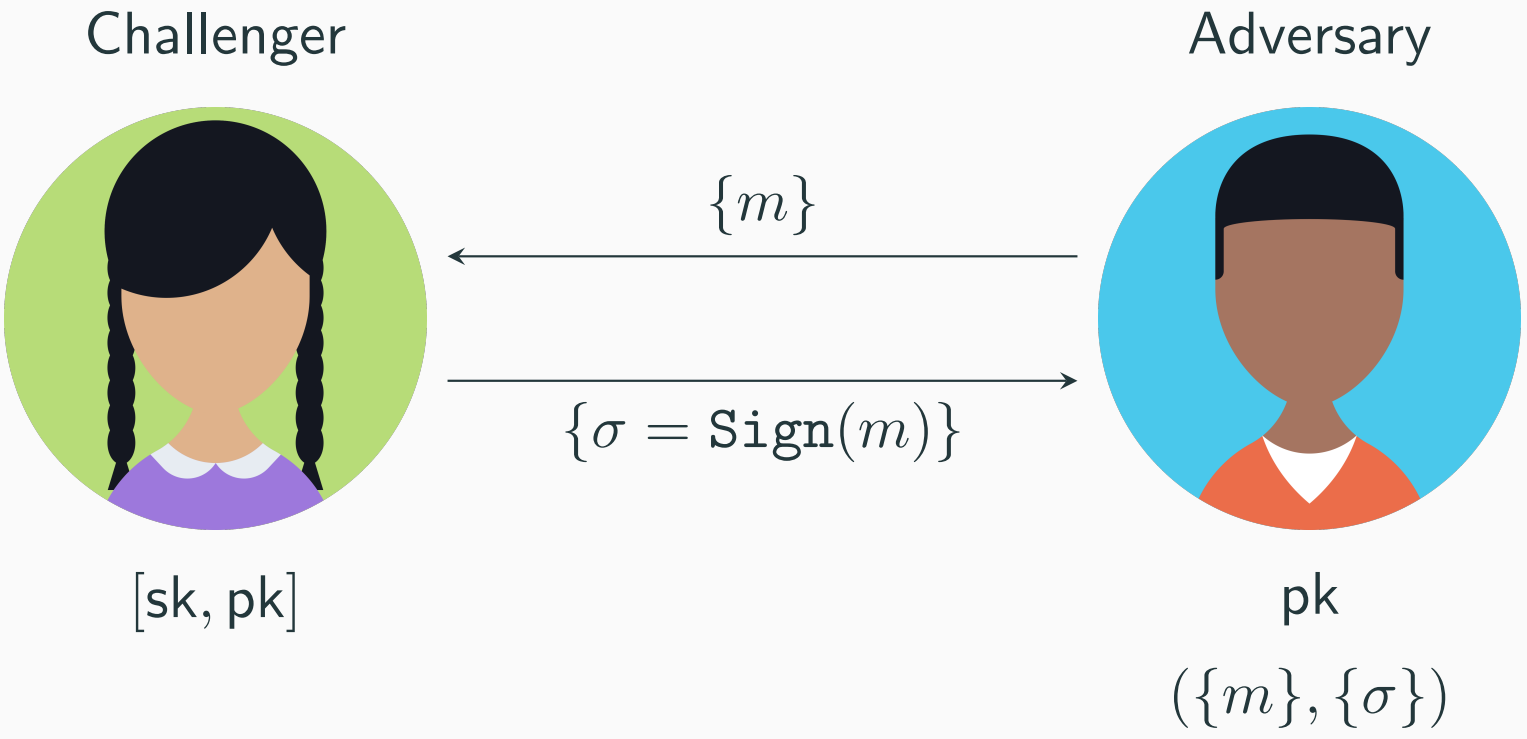
$[\mathsf{sk}, \mathsf{pk}]$

Adversary

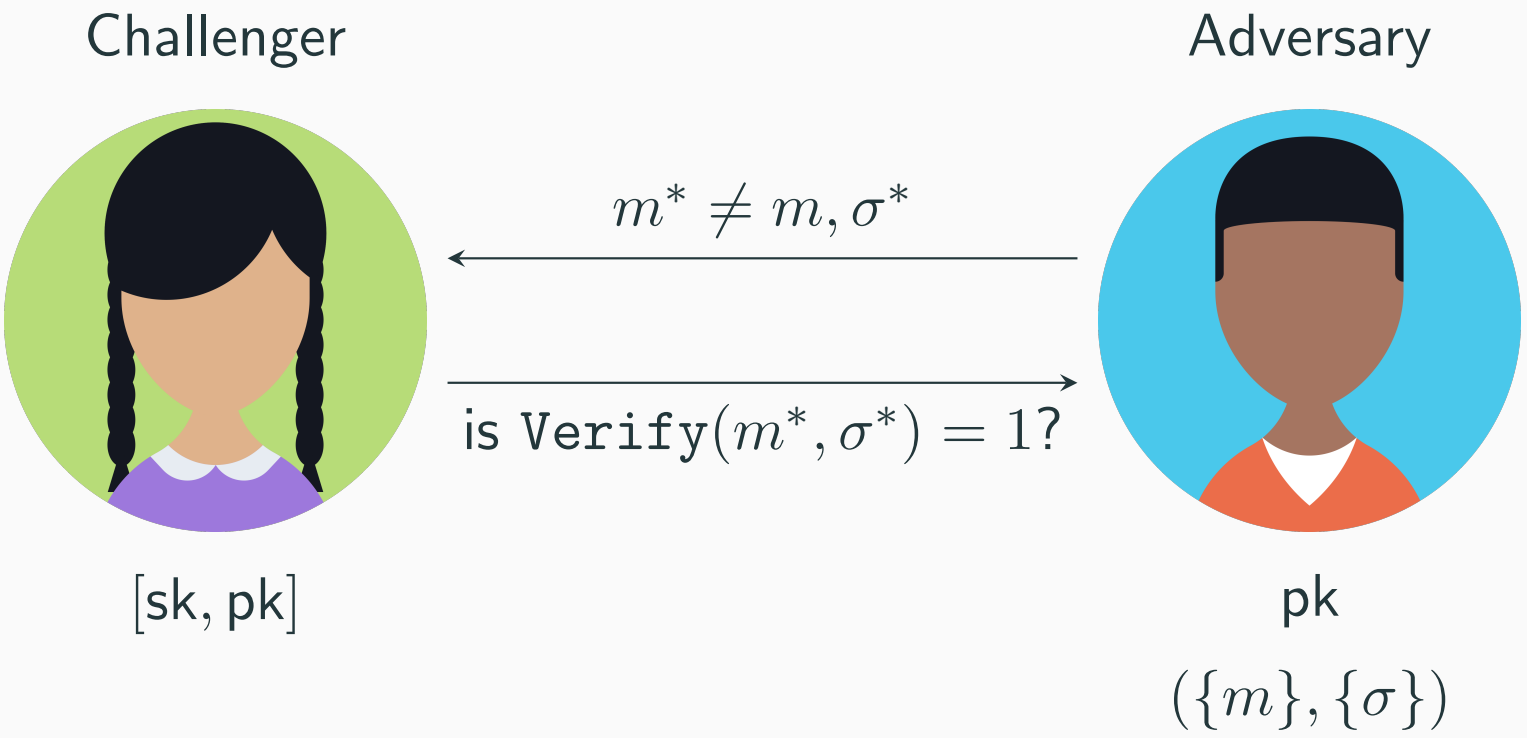
EUF-CMA [GMR88]



EUF-CMA [GMR88]



EUF-CMA [GMR88]



Zero-knowledge proof of knowledge

ZK proof of knowledge

Completeness: if the statement is true, an honest verifier will be convinced by an honest prover

Soundness: if the statement is false, no cheating prover can convince an honest verifier

Zero-knowledge: if the statement is true, no verifier learns anything other than the fact that the statement is true

Schnorr identification protocol (HVZK) [Sch90]

Verifier

public $y = g^x$

Prover

secret x

Schnorr identification protocol (HVZK) [Sch90]

Verifier

public $y = g^x$

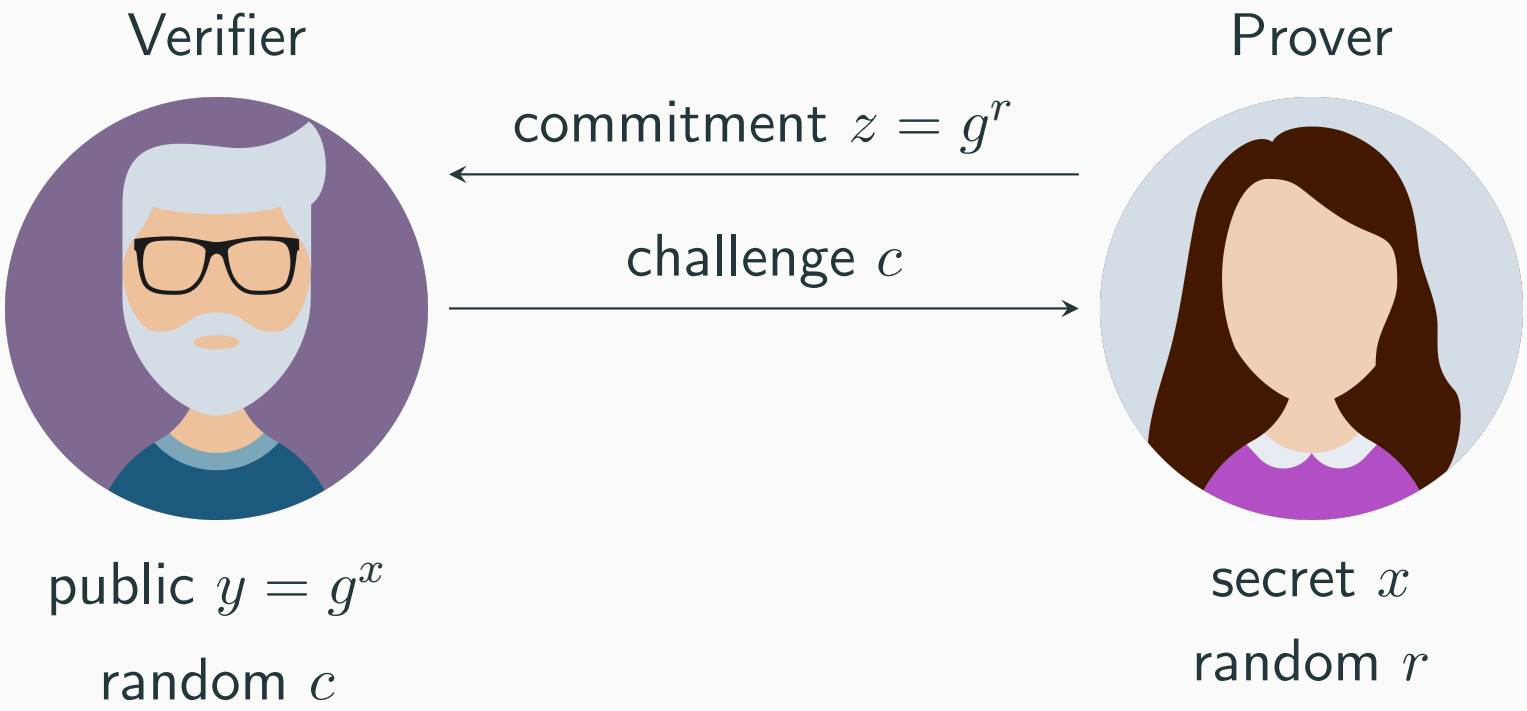
commitment $z = g^r$

Prover

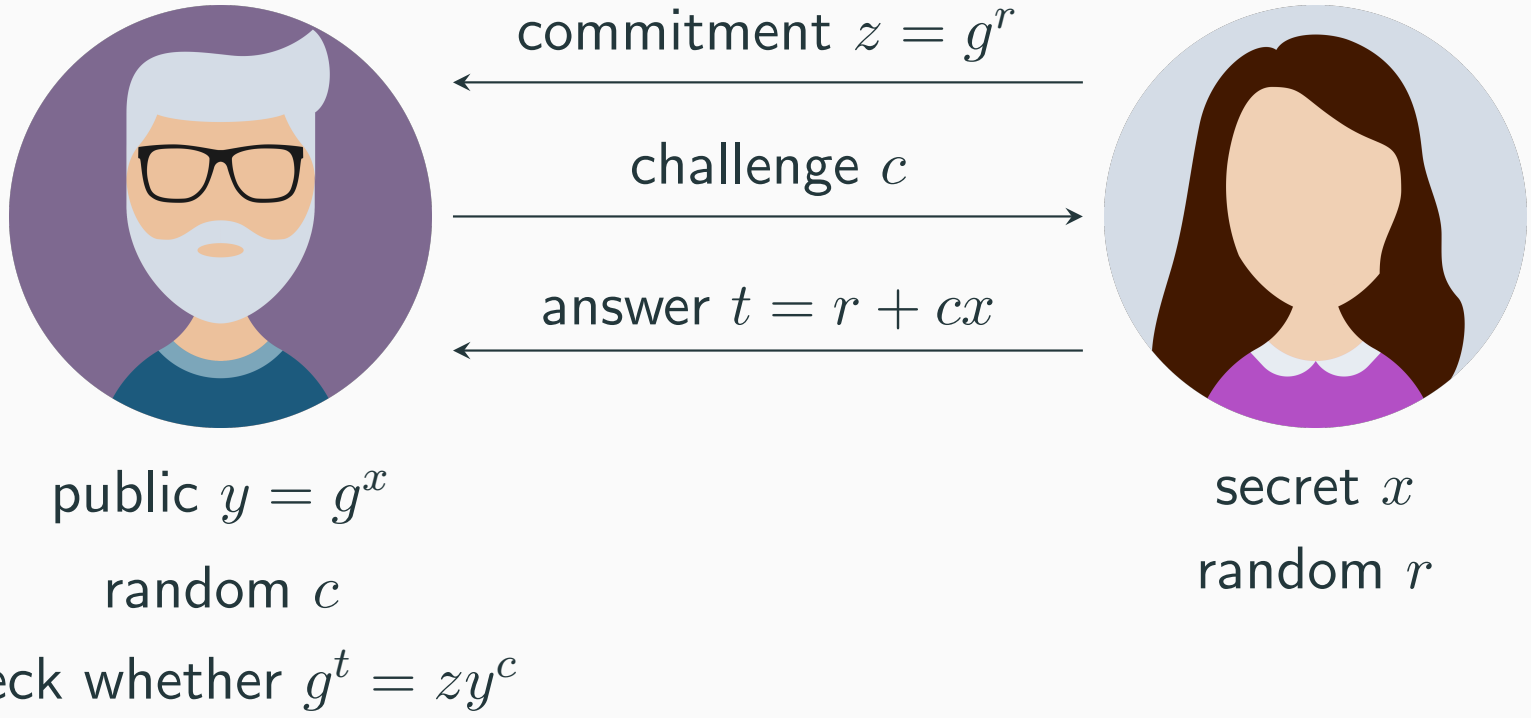
secret x

random r

Schnorr identification protocol (HVZK) [Sch90]



Schnorr identification protocol (HVZK) [Sch90]



Fiat-Shamir transform of Schnorr protocol (NIZK) [FS87]

Verifier

public $y = g^x$

Prover

secret x

Fiat-Shamir transform of Schnorr protocol (NIZK) [FS87]

Verifier

public $y = g^x$

Prover

secret x

commit $z = g^r$

Fiat-Shamir transform of Schnorr protocol (NIZK) [FS87]

Verifier

public $y = g^x$

Prover

secret x

commit $z = g^r$

challenge $c = \mathcal{H}(g, y, z)$

Fiat-Shamir transform of Schnorr protocol (NIZK) [FS87]

Verifier

public $y = g^x$

z, t

Prover

secret x

commit $z = g^r$

challenge $c = \mathcal{H}(g, y, z)$

answer $t = r + cx$

Fiat-Shamir transform of Schnorr protocol (NIZK) [FS87]

Verifier

public $y = g^x$

compute c

check whether $g^t = zy^c$

z, t

Prover

secret x

commit $z = g^r$

challenge $c = \mathcal{H}(g, y, z)$

answer $t = r + cx$

Signature of knowledge (SoK)

Verifier

message m

public $y = g^x$

Prover

message m

secret x

Signature of knowledge (SoK)

Verifier

message m

public $y = g^x$

Prover

message m

secret x

commit $z = g^r$

Signature of knowledge (SoK)

Verifier

message m

public $y = g^x$

Prover

message m

secret x

commit $z = g^r$

challenge $c = \mathcal{H}(g, y, z, m)$

Signature of knowledge (SoK)

Verifier

message m

public $y = g^x$

z, t

Prover

message m

secret x

commit $z = g^r$

challenge $c = \mathcal{H}(g, y, z, m)$

answer $t = r + cx$

Signature of knowledge (SoK)

Verifier

message m

public $y = g^x$

compute c

check whether $g^t = zy^c$

z, t

Prover

message m

secret x

commit $z = g^r$

challenge $c = \mathcal{H}(g, y, z, m)$

answer $t = r + cx$

Bilinear pairing [GPS08]

$\mathbb{G}_1 = \langle g \rangle$, $\mathbb{G}_2 = \langle \tilde{g} \rangle$ and $\mathbb{G}_T$ groups of order ℓ

map $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$

Bilinear: $e(g^a, \tilde{g}^b) = e(g, \tilde{g})^{ab}$

Non-degenerate: $e(g, \tilde{g}) \neq 1_{\mathbb{G}_T}$

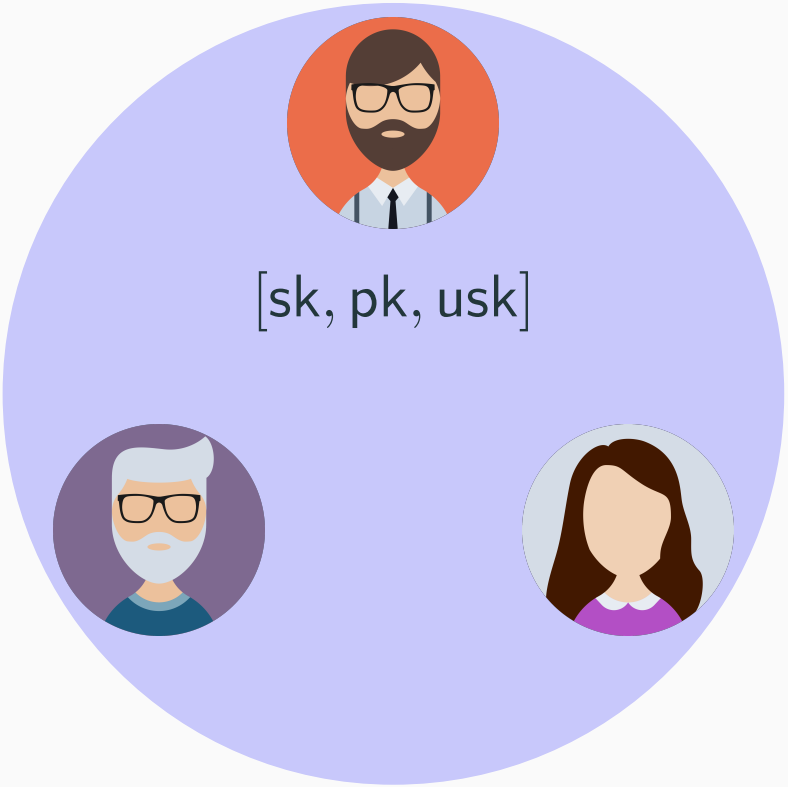
Computable: computable by a polynomial time algorithm

type-3: no easily computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ in either way

Introduction to group signatures

Idea: sketch [BSZ05]

Group Members



[sk, pk, usk]

Non-member

Group Manager

[gsk, gpk]

Opening Authority

[osk, opk]

Idea: security model [BMW03]

Correctness: each group member can produce valid signature

Anonymity: a valid signature cannot be tied to its issuer

Traceability: a valid signature must have been produce by a group member and its anonymity can be lifted

Non-frameability: no group member can be falsely accused of having produced a signature

Joining the group

Group Member

Group Manager

Interactive protocol

$[\mathsf{sk}, \mathsf{pk}]$

get certificate τ/usk on id

$[\mathsf{gsk}, \mathsf{gpk}]$

register user

SEP framework [BMW03]

Sign: $\sigma \leftarrow \Sigma.\mathtt{Sign}_{\mathsf{sk}}(m)$

Encrypt: $c \leftarrow \Gamma.\mathtt{Encrypt}_{\mathsf{opk}}(\sigma, \tau, \mathsf{pk})$

Prove: NIZK proof π that everything is well formed

Signature on m is (c, π)

Bichsel et al. [BCN+10]

Remove encryption

- Randomize certificate τ'
- SoK π on m that τ' certifies user

Signature on m is (τ', π)

PS signatures

Randomizable signature

PS signature [PS16]

- $\Sigma.\texttt{Setup}(1^\lambda) \to pp$

$$pp = (\mathbb{G}_1 = \langle g \rangle, \mathbb{G}_2 = \langle \tilde{g} \rangle, \mathbb{G}_T, e, X = g^x, \tilde{X} = \tilde{g}^x)$$

- $\Sigma.\texttt{Keygen}(pp) \to [\mathsf{sk} = g^y, \mathsf{pk} = \tilde{g}^y]$
- $\Sigma.\texttt{Sign}(\mathsf{sk}, m) \to (\sigma_1 = g^r, \sigma_2 = X^r(\mathsf{sk})^{rm})$
- $\Sigma.\texttt{Verify}(\mathsf{pk}, m, \sigma) \to$ is $e(\sigma_1, \tilde{X}(\mathsf{pk})^m) = e(\sigma_2, \tilde{g})$?

Same verification equation holds for (σ_1^s, σ_2^s)

A variant of PS signatures

Remember sk $= g^y$

$$\Sigma.\texttt{Sign}(g^y, m) \rightarrow (\sigma_1 = g^r, \sigma_2 = X^{r/m} g^{yr})$$

"Remove" $X^{r/m}$, then $(\sigma_1, \sigma_2) = (g^r, g^{yr})$ in same "projective equivalence-class" as (g, g^y)

FHS signatures

FHS signature on equivalence-class

Equivalence relation on $\mathbb{G}_1 \times \mathbb{G}_1$

$(m_1, m_2) \sim (n_1, n_2) \iff \exists r$ scalar, $\quad (m_1, m_2) = (n_1^r, n_2^r)$

FHS signature scheme [FHS19]

- $\Sigma.\mathtt{Setup}(1^\lambda) \to (\mathbb{G}_1 = \langle g \rangle, \mathbb{G}_2 = \langle \tilde{g} \rangle, \mathbb{G}_T, e)$
- $\Sigma.\mathtt{Keygen}(pp) \to [\mathsf{sk}, \mathsf{pk}]$

$$\mathsf{sk} = (\alpha_1, \alpha_2) \text{ scalars}, \quad \mathsf{pk} = (\tilde{A}_1, \tilde{A}_2) = (\tilde{g}^{\alpha_1}, \tilde{g}^{\alpha_2})$$

- $\Sigma.\mathtt{Sign}(\mathsf{sk}, (m_1, m_2)) \to (\tau_1, \tau_2, \tilde{\tau})$

$$\tau_1 = (m_1^{\alpha_1} m_2^{\alpha_2})^t, \quad \tau_2 = g^{1/t}, \quad \tilde{\tau} = \tilde{g}^{1/t}$$

- $\Sigma.\mathtt{Verify}(\mathsf{pk}, (m_1, m_2), (\tau_1, \tau_2, \tilde{\tau})) \to$ are

$$e(\tau_1, \tilde{\tau}) = e(m_1, \tilde{A}_1) e(m_2, \tilde{A}_2) \quad \text{and} \quad e(\tau_2, \tilde{g}) = e(g, \tilde{\tau})?$$

Randomizing FHS signatures

Reminder:

- $\Sigma.\texttt{Sign}(\mathsf{sk}, (m_1, m_2)) \to (\tau_1, \tau_2, \tilde{\tau})$

$$\tau_1 = (m_1^{\alpha_1} m_2^{\alpha_2})^t, \quad \tau_2 = g^{1/t}, \quad \tilde{\tau} = \tilde{g}^{1/t}$$

- $\Sigma.\texttt{Verify}(\mathsf{pk}, (m_1, m_2), (\tau_1, \tau_2, \tilde{\tau})) \to$ are

$$e(\tau_1, \tilde{\tau}) = e(m_1, \tilde{A}_1) e(m_2, \tilde{A}_2) \quad \text{and} \quad e(\tau_2, \tilde{g}) = e(g, \tilde{\tau})?$$

Pick $(m_1^r, m_2^r) \sim (m_1, m_2)$ and a random scalar t':

$$(\tau_1, \tau_2, \tilde{\tau}) \text{ signs } (m_1, m_2) \Rightarrow (\tau_1^{rt'}, \tau_2^{1/t'}, \tilde{\tau}^{1/t'}) \text{ signs } (m_1^r, m_2^r)$$

Our construction

Intuition: combining PS and FHS signatures

$\Sigma_{vPS}.\texttt{Sign}(g^y, m) \to (\sigma_1, \sigma_2) = (g^r, X^{r/m} g^{yr})$

- remove $X^{r/m}$: multiply σ_2 by $X^{-r/m}$
- so $(\sigma_1, X^{-r/m}\sigma_2) \sim (g, g^y)$
- choose certificate to be a FHS signature on (g, g^y)

$\Sigma_{FHS}.\texttt{Sign}(\mathsf{gsk}, (g, g^y)) \to (\tau_1, \tau_2, \tilde{\tau})$

Equations $\Sigma_{FHS}.\texttt{Verify}(\mathsf{gpk}, (g^r, g^{yr}), (\tau_1^r, \tau_2, \tilde{\tau}))$ are

$$e(\tau_1, \tilde{\tau}) = e(g^r, \tilde{A}_1) e(g^{yr}, \tilde{A}_2) \quad \text{and} \quad e(\tau_2, \tilde{g}) = e(g, \tilde{\tau})$$

on the public key $(\tilde{A}_1, \tilde{A}_2) = (\tilde{g}^{\alpha_1}, \tilde{g}^{\alpha_2})$

Intuition: combining PS and FHS signatures

How to *really* remove $X^{r/m}$ from $(\sigma_1, \sigma_2) = (g^r, X^{r/m} g^{yr})$?

$$
\begin{aligned}
e(\tau_1, \tilde{\tau}) &= e(g^r, \tilde{A}_1) e(g^{yr}, \tilde{A}_2) \\
&= e(\sigma_1, \tilde{A}_1) e(\sigma_2 X^{-r/m}, \tilde{A}_2) \\
&= e(\sigma_1, \tilde{A}_1) e(\sigma_2, \tilde{A}_2) e(g^{-rx/m}, \tilde{A}_2) \\
&= e(\sigma_1, \tilde{A}_1) e(\sigma_2, \tilde{A}_2) e(\sigma_1, \tilde{B}^{-1/m})
\end{aligned}
$$

Add $\tilde{B} = \tilde{X}^{\alpha_2}$ to FHS public key from the PS one

Our group signature scheme [CS18]

- $\texttt{Setup}(1^\lambda) \to pp$ (type-3 pairing and $X, \tilde{X}$)
- $\texttt{GKeygen}(pp) \to [\mathsf{gsk}, \mathsf{gpk}]$

  $\mathsf{gsk} = (\alpha_1, \alpha_2)$ scalars, $\quad \mathsf{gpk} = (\tilde{A}_1, \tilde{A}_2, \tilde{B}) = (\tilde{g}^{\alpha_1}, \tilde{g}^{\alpha_2}, \tilde{X}^{\alpha_2})$

- upon joining, user gets $(\tau_1, \tau_2, \tilde{\tau})$ certifying (g, g^y): set their group signing key to $\mathsf{usk} = (\tau_1, \tau_2, \tilde{\tau}, g^y)$
- Group Manager keeps $\texttt{Encrypt}_{\mathsf{opk}}(\tilde{g}^y)$
- $\texttt{Sign}(\mathsf{usk}, m) \to (\tau_1', \tau_2', \tilde{\tau}', \sigma_1, \sigma_2)$

  $(\tau_1', \tau_2', \tilde{\tau}') = (\tau_1^{rs}, \tau_2^{1/s}, \tilde{\tau}^{1/s}) \quad$ and $\quad (\sigma_1, \sigma_2) = (g^r, X^{r/m} g^{yr})$

- $\texttt{Verify}(\mathsf{gpk}, m, (\tau_1', \tau_2', \tilde{\tau}', \sigma_1, \sigma_2))$ checks whether

  $e(\tau_1', \tilde{\tau}') = e(\sigma_1, \tilde{A}_1 \tilde{B}^{-1/m}) e(\sigma_2, \tilde{A}_2) \quad$ and $\quad e(\tau_2', \tilde{g}) = e(g, \tilde{\tau}')$

Comparison

Comparison with other schemes

Scheme	Size	Cost	GS model	Anonymity
[BCN+10]	1664	3 e_1 + 1 e_T	BMW	selfless
[PS16]	1280	2 e_1 + 1 e_T	BMW	selfless
[DS18]	2816	5 e_1 + 1 e_2	BSZ	CPA
[DS18]*	4608	5 e_1 + 6 e_2	BSZ	full
[BHK+18]	4992	9 e_1 + 2 e_2	BMW	full
Ours	2304	5 e_1 + 1 e_2	BSZ	CPA & selfless
Ours*	2304	5 e_1 + 1 e_2	BMW	full

Table 1: Efficiency and security comparisons (see [CS18])

Thank you for your attention!

References

Patrik Bichsel, Jan Camenisch, Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. “Get Shorty via Group Signatures without Encryption”. In: *SCN 10*. Ed. by Juan A. Garay and Roberto De Prisco. Vol. 6280. LNCS. Springer, Heidelberg, Sept. 2010, pp. 381–398. DOI: 10.1007/978-3-642-15317-4_24.

Michael Backes, Lucjan Hanzlik, Kamil Kluczniak, and Jonas Schneider. "Signatures with Flexible Public Key: Introducing Equivalence Classes for Public Keys". In: *ASIACRYPT 2018, Part II*. Ed. by Thomas Peyrin and Steven Galbraith. Vol. 11273. LNCS. Springer, Heidelberg, Dec. 2018, pp. 405–434. DOI: 10.1007/978-3-030-03329-3_14.

Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. "Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions". In: *EUROCRYPT 2003*. Ed. by Eli Biham. Vol. 2656. LNCS. Springer, Heidelberg, May 2003, pp. 614–629. DOI: 10.1007/3-540-39200-9_38.

Mihir Bellare, Haixia Shi, and Chong Zhang. "Foundations of Group Signatures: The Case of Dynamic Groups". In: *CT-RSA 2005*. Ed. by Alfred Menezes. Vol. 3376. LNCS. Springer, Heidelberg, Feb. 2005, pp. 136–153. DOI: 10.1007/978-3-540-30574-3_11.

Rémi Clarisse and Olivier Sanders. *Short Group Signature without Random Oracles*. Cryptology ePrint Archive, Report 2018/1115. https://eprint.iacr.org/2018/1115. 2018.

David Derler and Daniel Slamanig. "Highly-Efficient Fully-Anonymous Dynamic Group Signatures". In: *ASIACCS 18*. Ed. by Jong Kim, Gail-Joon Ahn, Seungjoo Kim, Yongdae Kim, Javier López, and Taesoo Kim. ACM Press, Apr. 2018, pp. 551–565.

Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. "Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials". In: *Journal of Cryptology* 32.2 (Apr. 2019), pp. 498–546. DOI: 10.1007/s00145-018-9281-4.

Freepik. *All avatars icons made by Freepik from www.flaticon.com*. https://www.flaticon.com/authors/freepik.

Amos Fiat and Adi Shamir. "How to Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *CRYPTO'86*. Ed. by Andrew M. Odlyzko. Vol. 263. LNCS. Springer, Heidelberg, Aug. 1987, pp. 186–194. DOI: 10.1007/3-540-47721-7_12.

Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. "A Digital Signature Scheme Secure Against Adaptive Chosen-message Attacks". In: *SIAM Journal on Computing* 17.2 (Apr. 1988), pp. 281–308.

Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. "Pairings for cryptographers". In: *Discrete Applied Mathematics* 156.16 (2008), pp. 3113–3121. DOI: 10.1016/j.dam.2007.12.010. URL: https://doi.org/10.1016/j.dam.2007.12.010.

David Pointcheval and Olivier Sanders. "Short Randomizable Signatures". In: *CT-RSA 2016*. Ed. by Kazue Sako. Vol. 9610. LNCS. Springer, Heidelberg, Feb. 2016, pp. 111–126. DOI: 10.1007/978-3-319-29485-8_7.

Claus-Peter Schnorr. “Efficient Identification and Signatures for Smart Cards”. In: *CRYPTO’89*. Ed. by Gilles Brassard. Vol. 435. LNCS. Springer, Heidelberg, Aug. 1990, pp. 239–252. DOI: 10.1007/0-387-34805-0_22.