Generic Plaintext Equality and Inequality Proofs

Olivier Blazy [1] Xavier Bultel [2] Pascal Lafourcade [3]
Octavio Perez Kempner [4,5]

[1]Université de Limoges, XLIM, Limoges, France

[2]INSA Centre Val de Loire, LIFO lab, France

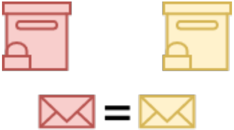
[3]University Clermont Auvergne, LIMOS, France

[4]DIENS, École normale supérieure, CNRS, PSL University, Paris, France

[5]be-ys Research, France

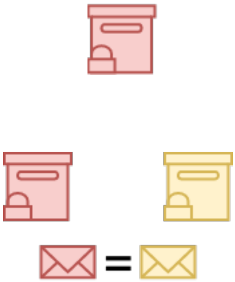
Agenda

1. Motivation
2. Generic Randomizable Encryption
3. Protocols
4. Comparisons for ElGamal
5. Conclusions and Future Work

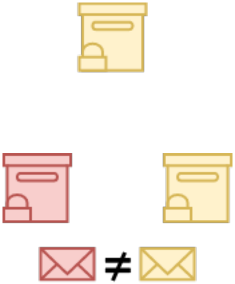


Plaintext Equality (PET)

Motivation

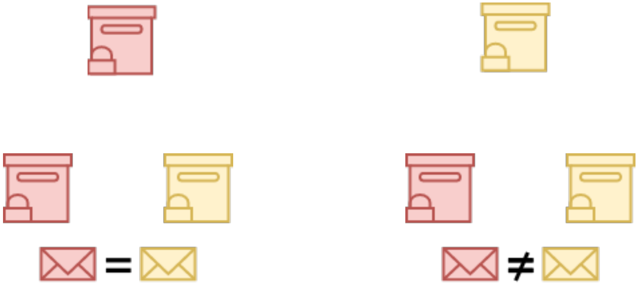


Plaintext Equality (PET)



Plaintext Inequality (PIT)

Motivation



Plaintext Equality (PET) Plaintext Inequality (PIT)

Generic zero knowledge proofs for PET-PIT

Applications

Applications

Voting

Voting

Reputation systems

Applications

Voting

Reputation systems

Cloud applications

Voting

Reputation systems

Cloud applications

Broadcast

Voting

Reputation systems

Cloud applications

Broadcast

Storage

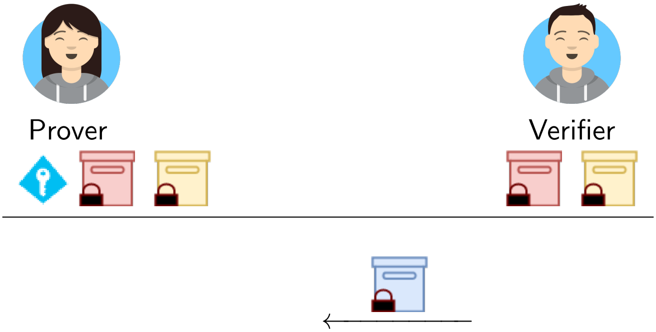
Warm Up Example

Prover

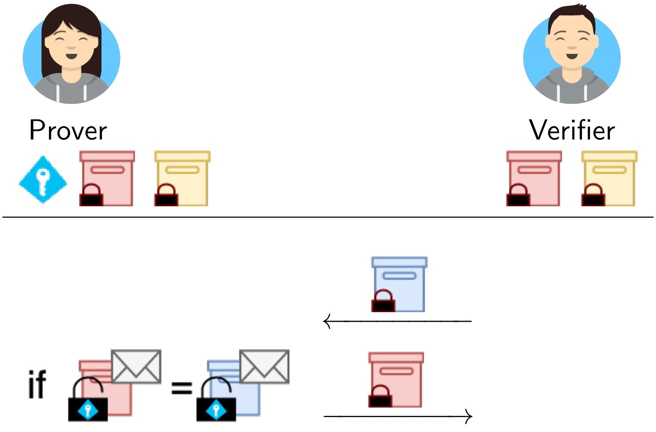
Verifier

Warm Up Example

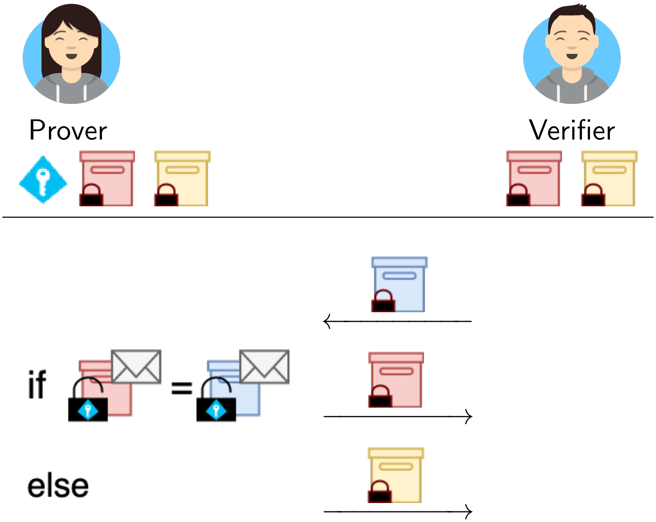
Warm Up Example



Warm Up Example



Warm Up Example



Agenda

Generic Randomizable Encryption

- Ciphertexts (Rand),

- Ciphertexts (Rand), messages (MsgRand),

- Ciphertexts (Rand), messages (MsgRand), encryption keys (KeyRand)

- Ciphertexts (Rand), messages (MsgRand), encryption keys (KeyRand) and combinations

Generic Randomizable Encryption

- Ciphertexts (Rand), messages (MsgRand), encryption keys (KeyRand) and combinations
- Formal definitions: randomizabily and strong randomizability, message-randomizability, key-randomizability and random coin decryption (RCD)

Generic Randomizable Encryption

- Ciphertexts (Rand), messages (MsgRand), encryption keys (KeyRand) and combinations
- Formal definitions: randomizabily and strong randomizability, message-randomizability, key-randomizability and random coin decryption (RCD)
- Two flavours: computational and perfect

Agenda

Protocols

Protocols

- Simple cut-and-choose protocols

Protocols

- Simple cut-and-choose protocols
- Completeness, soundness and perfect zero knowledge

Protocols

- Simple cut-and-choose protocols
- Completeness, soundness and perfect zero knowledge
- PIT: Rand

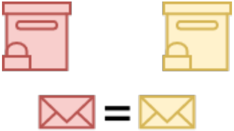
Protocols

- Simple cut-and-choose protocols
- Completeness, soundness and perfect zero knowledge
- PIT: Rand
- PET: Rand & MsgRand

Protocols

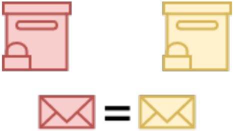
- Simple cut-and-choose protocols
- Completeness, soundness and perfect zero knowledge
- PIT: Rand
- PET: Rand & MsgRand
- Sigma PET's: Rand, MsgRand & (KeyRand $\vee$ RCD)

Protocols



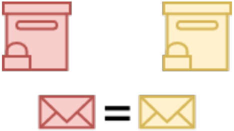
PET

Protocols



PET

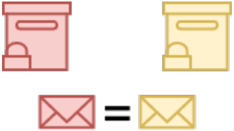
- $\mathsf{pk}_0 = \mathsf{pk}_1$ and the prover knows sk_0



PET

- $\mathsf{pk}_0 = \mathsf{pk}_1$ and the prover knows sk_0
  $\rightsquigarrow$ HPEQ, PEQ

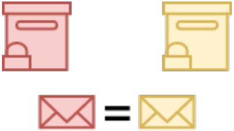
Protocols



PET

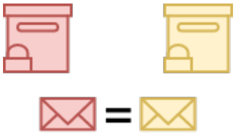
- $\mathsf{pk}_0 = \mathsf{pk}_1$ and the prover knows sk_0
  $\rightsquigarrow$ HPEQ, PEQ
- $\mathsf{pk}_0 \neq \mathsf{pk}_1$ and the prover knows sk_0 and sk_1

Protocols



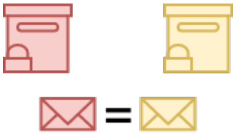
PET

- $pk_0 = pk_1$ and the prover knows sk_0
  $\rightsquigarrow$ HPEQ, PEQ
- $pk_0 \neq pk_1$ and the prover knows sk_0 and sk_1
  $\rightsquigarrow$ MATCHPEQ, SIGPEQ



PET

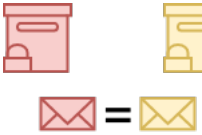
- $pk_0 = pk_1$ and the prover knows sk_0
  $\rightsquigarrow$ HPEQ, PEQ
- $pk_0 \neq pk_1$ and the prover knows sk_0 and sk_1
  $\rightsquigarrow$ MATCHPEQ, SIGPEQ
- $pk_0 \neq pk_1$ and the prover knows r_0 and r_1



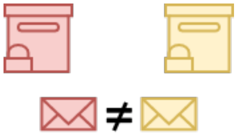
PET

- $\mathsf{pk}_0 = \mathsf{pk}_1$ and the prover knows sk_0
  $\rightsquigarrow$ HPEQ, PEQ
- $\mathsf{pk}_0 \neq \mathsf{pk}_1$ and the prover knows sk_0 and sk_1
  $\rightsquigarrow$ MATCHPEQ, SIGPEQ
- $\mathsf{pk}_0 \neq \mathsf{pk}_1$ and the prover knows r_0 and r_1
  $\rightsquigarrow$ RSPEQ

Protocols



PET



PIT

- $pk_0 = pk_1$ and the prover knows sk_0
  $\rightsquigarrow$ HPEQ, PEQ
- $pk_0 \neq pk_1$ and the prover knows sk_0 and sk_1
  $\rightsquigarrow$ MATCHPEQ, SIGPEQ
- $pk_0 \neq pk_1$ and the prover knows r_0 and r_1
  $\rightsquigarrow$ RSPEQ

- $pk_0 = pk_1$ and the prover knows sk_0
  $\rightsquigarrow$ HPINEQ, PINEQ

Alice $(\mathsf{sk}, \mathsf{pk}, c_0, c_1)$

Bob (pk, c_0, c_1)

HPINEQ

Alice $(\mathsf{sk}, \mathsf{pk}, c_0, c_1)$

Bob (pk, c_0, c_1)

$r \xleftarrow{\$} \mathcal{R};$

HPINEQ

Alice $(\mathsf{sk}, \mathsf{pk}, c_0, c_1)$

Bob (pk, c_0, c_1)

$r \xleftarrow{\$} \mathcal{R};\ b \xleftarrow{\$} \{0, 1\}$

HPINEQ

Alice $(\mathsf{sk}, \mathsf{pk}, c_0, c_1)$

Bob (pk, c_0, c_1)

$r \xleftarrow{\$} \mathcal{R};\ b \xleftarrow{\$} \{0, 1\}$

$c'_b \leftarrow \mathsf{Rand}(c_b, r)$

$\xleftarrow{\quad c'_b \quad}$

if $\mathsf{Dec}_{\mathsf{sk}}(c'_b) = \mathsf{Dec}_{\mathsf{sk}}(c_0)$

Alice $(\mathsf{sk}, \mathsf{pk}, c_0, c_1)$

Bob (pk, c_0, c_1)

Alice		Bob
		$r \xleftarrow{\$} \mathcal{R};\ b \xleftarrow{\$} \{0,1\}$
if $\mathsf{Dec}_{\mathsf{sk}}(c'_b) = \mathsf{Dec}_{\mathsf{sk}}(c_0)$	$\xleftarrow{\quad c'_b \quad}$	$c'_b \leftarrow \mathsf{Rand}(c_b, r)$
then $z = 0$ **else** $z = 1$	$\xrightarrow{\quad z \quad}$	**if** $(z = b)$ **then** Accept **else** Reject

Alice (sk, pk, c_0, c_1)		**Bob** (pk, c_0, c_1)
		$r \xleftarrow{\$} \mathcal{R};\ b \xleftarrow{\$} \{0,1\}$
if $\mathsf{Dec}_{\mathsf{sk}}(c'_b) = \mathsf{Dec}_{\mathsf{sk}}(c_0)$	$\xleftarrow{\quad c'_b \quad}$	$c'_b \leftarrow \mathsf{Rand}(c_b, r)$
then $z = 0$ **else** $z = 1$	$\xrightarrow{\quad z \quad}$	**if** $(z = b)$ **then** Accept **else** Reject

Theorem

If the PKE scheme is (computationally) randomizable, then HPINEQ *is complete, computationally sound and perfect HVZK.*

HPEQ

Alice (sk, pk, c_0, c_1)

Bob (pk, c_0, c_1)

HPEQ

Alice (sk, pk, c_0, c_1)

Bob (pk, c_0, c_1)

$r \xleftarrow{\$} \mathcal{R};$

HPEQ

Alice $(\mathsf{sk}, \mathsf{pk}, c_0, c_1)$

Bob (pk, c_0, c_1)

$r \xleftarrow{\$} \mathcal{R}; r_{\mathsf{m}} \xleftarrow{\$} \mathcal{R}_{\mathsf{M}};$

HPEQ

Alice $(\mathsf{sk}, \mathsf{pk}, c_0, c_1)$

Bob (pk, c_0, c_1)

$r \xleftarrow{\$} \mathcal{R}; r_{\mathsf{m}} \xleftarrow{\$} \mathcal{R}_{\mathsf{M}};\ b \xleftarrow{\$} \{0, 1\}$

HPEQ

Alice $(\mathsf{sk}, \mathsf{pk}, c_0, c_1)$

Bob (pk, c_0, c_1)

$r \xleftarrow{\$} \mathcal{R}; r_{\mathrm{m}} \xleftarrow{\$} \mathcal{R}_{\mathrm{M}};\ b \xleftarrow{\$} \{0, 1\}$

$c'_b \leftarrow \mathsf{Rand}(c_b, r)$

HPEQ

Alice $(\mathsf{sk}, \mathsf{pk}, c_0, c_1)$		**Bob** (pk, c_0, c_1)
		$r \xleftarrow{\$} \mathcal{R}; r_\mathsf{m} \xleftarrow{\$} \mathcal{R}_\mathsf{M};\ b \xleftarrow{\$} \{0,1\}$
		$c'_b \leftarrow \mathsf{Rand}(c_b, r)$
$m' \leftarrow \mathsf{Dec}_\mathsf{sk}(c''_b); m \leftarrow \mathsf{Dec}_\mathsf{sk}(c_0)$	$\xleftarrow{\quad c''_b \quad}$	$c''_b \leftarrow \mathsf{MsgRandC}(c'_b, r_\mathsf{m})$

HPEQ

Alice (sk, pk, c_0, c_1)

Bob (pk, c_0, c_1)

Alice		Bob
		$r \xleftarrow{\$} \mathcal{R}; r_{\mathrm{m}} \xleftarrow{\$} \mathcal{R}_{\mathrm{M}};\ b \xleftarrow{\$} \{0,1\}$
		$c'_b \leftarrow \mathsf{Rand}(c_b, r)$
$m' \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c''_b);\ m \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c_0)$	$\xleftarrow{\quad c''_b \quad}$	$c''_b \leftarrow \mathsf{MsgRandC}(c'_b, r_{\mathrm{m}})$
$z \leftarrow \mathsf{MsgRandExt}(m', m)$	$\xrightarrow{\quad z \quad}$	**if** $(z = r_{\mathrm{m}})$ **then** Accept **else** Reject

Alice $(\mathsf{sk}, \mathsf{pk}, c_0, c_1)$		**Bob** (pk, c_0, c_1)
		$r \xleftarrow{\$} \mathcal{R};\ r_{\mathsf{m}} \xleftarrow{\$} \mathcal{R}_{\mathsf{M}};\ b \xleftarrow{\$} \{0,1\}$
		$c'_b \leftarrow \mathsf{Rand}(c_b, r)$
$m' \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c''_b);\ m \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c_0)$	$\xleftarrow{\quad c''_b \quad}$	$c''_b \leftarrow \mathsf{MsgRandC}(c'_b, r_{\mathsf{m}})$
$z \leftarrow \mathsf{MsgRandExt}(m', m)$	$\xrightarrow{\quad z \quad}$	**if** $(z = r_{\mathsf{m}})$ **then** Accept **else** Reject

Theorem

If the PKE scheme is (computationally) randomizable, (computationally) message-randomizable and message-random-extractable, then HPEQ *is complete, computationally sound and perfect HVZK.*

Alice $(r_1, r_2, \mathsf{pk}_1, \mathsf{pk}_2, c_1, c_2)$ — **Bob** $\mathsf{V}(\mathsf{pk}_1, \mathsf{pk}_2, c_1, c_2)$

$r_{\mathsf{m}} \xleftarrow{\$} \mathcal{R}_{\mathsf{M}}$; $(r_1', r_2') \xleftarrow{\$} \mathcal{R}^2$

$r_1'' \leftarrow \mathsf{RandR}(r_1, r_1')$; $r_2'' \leftarrow \mathsf{RandR}(r_2, r_2')$

$c_1' \leftarrow \mathsf{Rand}(c_1, r_1')$; $c_2' \leftarrow \mathsf{Rand}(c_2, r_2')$

$c_1'' \leftarrow \mathsf{MsgRandC}(c_1', r_{\mathsf{m}})$

$c_2'' \leftarrow \mathsf{MsgRandC}(c_2', r_{\mathsf{m}})$ $\xrightarrow{(c_1'', c_2'')}$

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

if $(b = 0)$ **then** $z = (r_1'', r_2'')$ $\xrightarrow{z}$ **if** $b = 0$ **then return** $(\mathsf{CDec}_{r_1''}(c_1'', \mathsf{pk}_1) = \mathsf{CDec}_{r_2''}(c_2'', \mathsf{pk}_2))$

else $z = (r_1', r_2', r_{\mathsf{m}})$

else $\widetilde{c}_1' \leftarrow \mathsf{Rand}(c_1, r_1')$; $\widetilde{c}_2' \leftarrow \mathsf{Rand}(c_2, r_2')$;

$\widetilde{c}_1'' \leftarrow \mathsf{MsgRandC}(\widetilde{c}_1', r_{\mathsf{m}})$; $\widetilde{c}_2'' \leftarrow \mathsf{MsgRandC}(\widetilde{c}_2', r_{\mathsf{m}})$

return $((\widetilde{c}_1'' = c_1'') \wedge (\widetilde{c}_2'' = c_2''))$

Alice $(r_1, r_2, \mathsf{pk}_1, \mathsf{pk}_2, c_1, c_2)$ — **Bob** $\mathsf{V}(\mathsf{pk}_1, \mathsf{pk}_2, c_1, c_2)$

Alice:

$r_\mathrm{m} \xleftarrow{\$} \mathcal{R}_\mathrm{M}$; $(r'_1, r'_2) \xleftarrow{\$} \mathcal{R}^2$

$r''_1 \leftarrow \mathsf{RandR}(r_1, r'_1)$; $r''_2 \leftarrow \mathsf{RandR}(r_2, r'_2)$

$c'_1 \leftarrow \mathsf{Rand}(c_1, r'_1)$; $c'_2 \leftarrow \mathsf{Rand}(c_2, r'_2)$

$c''_1 \leftarrow \mathsf{MsgRandC}(c'_1, r_\mathrm{m})$

$c''_2 \leftarrow \mathsf{MsgRandC}(c'_2, r_\mathrm{m})$

Alice → Bob: $\xrightarrow{(c''_1, c''_2)}$

Bob: $b \xleftarrow{\$} \{0, 1\}$

Bob → Alice: $\xleftarrow{b}$

Alice:

if $(b = 0)$ **then** $z = (r''_1, r''_2)$

else $z = (r'_1, r'_2, r_\mathrm{m})$

Alice → Bob: $\xrightarrow{z}$

Bob:

if $b = 0$ **then return** $(\mathsf{CDec}_{r''_1}(c''_1, \mathsf{pk}_1) = \mathsf{CDec}_{r''_2}(c''_2, \mathsf{pk}_2))$

else $\widetilde{c}'_1 \leftarrow \mathsf{Rand}(c_1, r'_1)$; $\widetilde{c}'_2 \leftarrow \mathsf{Rand}(c_2, r'_2)$;

$\widetilde{c}''_1 \leftarrow \mathsf{MsgRandC}(\widetilde{c}'_1, r_\mathrm{m})$; $\widetilde{c}''_2 \leftarrow \mathsf{MsgRandC}(\widetilde{c}'_2, r_\mathrm{m})$

return $((\widetilde{c}''_1 = c''_1) \wedge (\widetilde{c}''_2 = c''_2))$

Theorem

If the PKE scheme is perfectly strong randomizable, random-extractable, perfectly message-randomizable and RCD*, then* RSPEQ *is complete, special sound, and perfect zero-knowledge.*

Protocols' Compatibility

Protocols' Compatibility

						Perfect ZK		ZKPoK		
Scheme	Security	RCD	Rand	MsgRand	KeyRand	PEQ	PINEQ	MATCHPEQ	SIGPEQ	RSPEQ
ElGamal [ElG85]	IND-CPA	✓	✓	✓	✓	✓	✓	✓	✓	✓
Paillier [Pai99]	IND-CPA	✓	✓	✓		✓	✓	✓		✓
GM [GM82]	IND-CPA		✓	✓		✓	✓	✓		
DEG [Dam91]	IND-CCA1	✓	✓	✓	✓	✓	✓	✓	✓	✓
CS-lite [CS98]	IND-CCA1	✓	✓	✓		✓	✓			✓
DSCS [PR07]	RCCA	✓	✓				✓			

Agenda

Comparisons for ElGamal

	PET			PIT	
Protocol	[CP93]	PEQ	RSPEQ	[CS03]	PINEQ
Prover	2EXP	6EXP	4EXP	6EXP	6EXP
Verifier	2EXP	4EXP	4EXP	4EXP	4EXP
Rounds	3	4	3	3	4

Comparisons for ElGamal

Protocol	HPEQ	PEQ	HPINEQ	PINEQ	RSPEQ	SIGPEQ
Avg. time (ms)	27.47	70.31	26.13	68.75	62.12	112.98
Deviation	0.21	1.28	0.15	0.6	2.06	3.70

Agenda

Conclusions

Conclusions

- Formal definitions for randomizability properties

Conclusions

- Formal definitions for randomizability properties
- Intuitive constructions of zero-knowledge PET-PIT protocols

Conclusions

- Formal definitions for randomizability properties
- Intuitive constructions of zero-knowledge PET-PIT protocols
- Non-interactive variants for sigma protocols via Fiat-Shamir

Conclusions

- Formal definitions for randomizability properties
- Intuitive constructions of zero-knowledge PET-PIT protocols
- Non-interactive variants for sigma protocols via Fiat-Shamir
- Applicable to real-world problems in a “plug & play” manner

Future Work

Future Work

- Design non-interactive protocols for plaintext inequality

Future Work

- Design non-interactive protocols for plaintext inequality
- Improve the number of rounds

Future Work

- Design non-interactive protocols for plaintext inequality
- Improve the number of rounds
- Build generic plaintext inequality tests $(<, \leq, \geq, >)$

Future Work

- Design non-interactive protocols for plaintext inequality
- Improve the number of rounds
- Build generic plaintext inequality tests $(<, \leq, \geq, >)$

Thank you for your time!

- David Chaum and Torben Pryds Pedersen.
  Wallet Databases with Observers.
  In Ernest F. Brickell, editor, *Advances in Cryptology — CRYPTO' 92*, pages 89–105. Springer Berlin Heidelberg, 1993.

- Ronald Cramer and Victor Shoup.
  A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack.
  In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, pages 13–25. Springer Berlin Heidelberg, 1998.

- Jan Camenisch and Victor Shoup.
  Practical Verifiable Encryption and Decryption of Discrete Logarithms.
  In *CRYPTO 2003*. Springer, 2003.

Ivan Damgård.
Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks.
In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, page 445–456. Springer-Verlag, 1991.

Taher ElGamal.
A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms.
In George Robert Blakley and David Chaum, editors, *Advances in Cryptology*, pages 10–18. Springer Berlin Heidelberg, 1985.

References III

Shafi Goldwasser and Silvio Micali.
Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information.
In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 365–377, New York, NY, USA, 1982. Association for Computing Machinery.

Pascal Paillier.
Public-Key Cryptosystems Based on Composite Degree Residuosity Classes.
In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 223–238. Springer Berlin Heidelberg, 1999.

References IV

Manoj Prabhakaran and Mike Rosulek.
Rerandomizable RCCA Encryption.
In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, pages 517–534. Springer Berlin Heidelberg, 2007.