Random Oracles in a Quantum World

Dan Boneh[1] Özgür Dagdelen[2] Marc Fischlin[2]
Anja Lehmann[3] Christian Schaffner[4] Mark Zhandry[1]

[1]Stanford University, USA

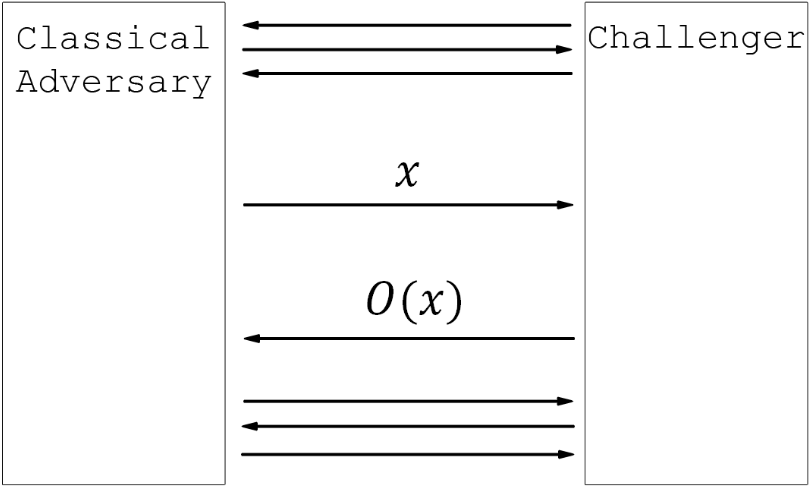
[2]CASED & Darmstadt University of Technology, Germany

[3]IBM Research Zurich, Switzerland

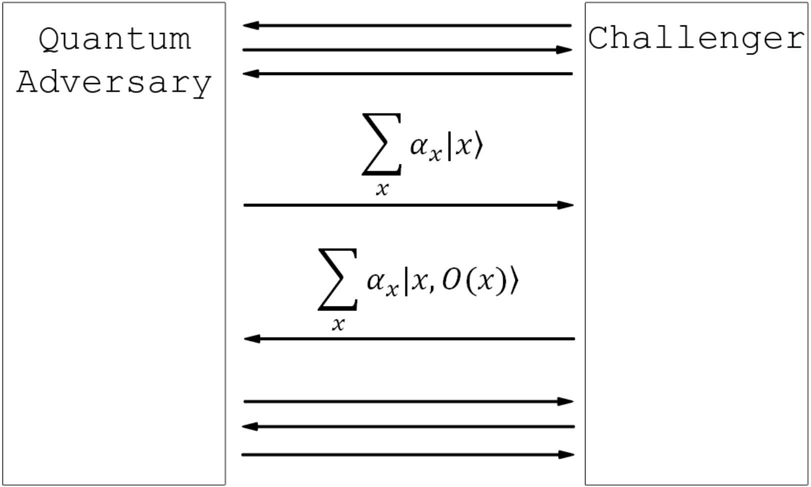
[4]University of Amsterdam and CWI, The Netherlands

December 5, 2011

Classical Random Oracle Model Adversaries



Quantum Random Oracle Model Adversaries



Quantum Random Oracle Model (QROM)

- Why quantum queries? Random oracle models hash function, which a quantum adversary can evaluate on superposition.

Quantum Random Oracle Model (QROM)

- Why quantum queries? Random oracle models hash function, which a quantum adversary can evaluate on superposition.
- Because quantum adversaries can query on a superposition, classical proofs of security do not carry over to the quantum setting.

Quantum Random Oracle Model (QROM)

- Why quantum queries? Random oracle models hash function, which a quantum adversary can evaluate on superposition.
- Because quantum adversaries can query on a superposition, classical proofs of security do not carry over to the quantum setting.
  Examples:
  - Simulating the random oracle
  - Determining what points the adversary is interested in
  - Programming the random oracle
  - Rewinding

Our Results

- Separation result: Scheme secure in classical ROM, but insecure in QROM

Our Results

- Separation result: Scheme secure in classical ROM, but insecure in QROM
  - Identification scheme

Our Results

- Separation result: Scheme secure in classical ROM, but insecure in QROM
    - Identification scheme
- Positive result: Signature Schemes

Our Results

- Separation result: Scheme secure in classical ROM, but insecure in QROM
  - Identification scheme
- Positive result: Signature Schemes
  - Some classical security proofs carry over (if quantum PRFs exist).

Our Results

- Separation result: Scheme secure in classical ROM, but insecure in QROM
  - Identification scheme
- Positive result: Signature Schemes
  - Some classical security proofs carry over (if quantum PRFs exist).
  - Example: Lattice-based signatures ([GPV08])
  - Example: Specific instances of Full Domain Hash
  - Generic Full Domain Hash is still open.

Our Results

- Separation result: Scheme secure in classical ROM, but insecure in QROM
  - Identification scheme
- Positive result: Signature Schemes
  - Some classical security proofs carry over (if quantum PRFs exist).
  - Example: Lattice-based signatures ([GPV08])
  - Example: Specific instances of Full Domain Hash
  - Generic Full Domain Hash is still open.
- Positive result: Encryption Schemes

Preimage Sampleable Functions

- A preimage sampleable trapdoor function (PSF) $\mathcal{F}$ is a triple of functions (G, f, f^{-1}):
  - $G(1^n)$ outputs $(\mathtt{sk}, \mathtt{pk})$
  - $f_{\mathtt{pk}}(x)$ is efficiently computable, uniformly distributed for random x.
  - $f_{\mathtt{sk}}^{-1}(y)$ samples uniformly from the set of x such that $f_{\mathtt{pk}}(x) = y$

Preimage Sampleable Functions

- A preimage sampleable trapdoor function (PSF) $\mathcal{F}$ is a triple of functions (G, f, f^{-1}):
  - $G(1^n)$ outputs $(\mathtt{sk}, \mathtt{pk})$
  - $f_{\mathtt{pk}}(x)$ is efficiently computable, uniformly distributed for random x.
  - $f_{\mathtt{sk}}^{-1}(y)$ samples uniformly from the set of x such that $f_{\mathtt{pk}}(x) = y$
- $\mathcal{F} = (G, f, f^{-1})$ is secure if it is one-way, collision-resistant, and has high preimage min-entropy.

Preimage Sampleable Functions

- A preimage sampleable trapdoor function (PSF) $\mathcal{F}$ is a triple of functions (G, f, f^{-1}):
  - $G(1^n)$ outputs $(\mathtt{sk}, \mathtt{pk})$
  - $f_{\mathtt{pk}}(x)$ is efficiently computable, uniformly distributed for random x.
  - $f_{\mathtt{sk}}^{-1}(y)$ samples uniformly from the set of x such that $f_{\mathtt{pk}}(x) = y$
- $\mathcal{F} = (G, f, f^{-1})$ is secure if it is one-way, collision-resistant, and has high preimage min-entropy.
- Secure construction from lattices [GPV08]

Example: GPV Signatures

Given a PSF $\mathcal{F} = (G, f, f^{-1})$, construct a signature scheme $\mathcal{S}^O = (G, S^O, V^O)$ as follows:

Example: GPV Signatures

Given a PSF $\mathcal{F} = (G, f, f^{-1})$, construct a signature scheme $\mathcal{S}^O = (G, S^O, V^O)$ as follows:

- $S^O_{\mathtt{sk}}(m) = f^{-1}_{\mathtt{sk}}(O(m))$. Remember this output for future queries of m

Example: GPV Signatures

Given a PSF $\mathcal{F} = (G, f, f^{-1})$, construct a signature scheme $\mathcal{S}^O = (G, S^O, V^O)$ as follows:

- $S^O_{\mathrm{sk}}(m) = f^{-1}_{\mathrm{sk}}(O(m))$. Remember this output for future queries of m
- $V^O_{\mathrm{pk}}(m, \sigma)$ accepts if and only if $f_{\mathrm{pk}}(\sigma) = O(m)$.

Example: GPV Signatures

Given a PSF $\mathcal{F} = (G, f, f^{-1})$, construct a signature scheme $\mathcal{S}^O = (G, S^O, V^O)$ as follows:

- $S^O_{\mathrm{sk}}(m) = f^{-1}_{\mathrm{sk}}(O(m))$. Remember this output for future queries of m
- $V^O_{\mathrm{pk}}(m, \sigma)$ accepts if and only if $f_{\mathrm{pk}}(\sigma) = O(m)$.

Theorem

Suppose $\mathcal{F}$ is a quantum-secure PSF, and that quantum pseudorandom functions exist. Then $\mathcal{S}$ is quantum secure.

Security of GPV Signatures

Two parts:

Security of GPV Signatures

Two parts:

- Prove that security of a certain type of classical reduction (called *history free*) implies security in the quantum setting

Security of GPV Signatures

Two parts:

- Prove that security of a certain type of classical reduction (called *history free*) implies security in the quantum setting
- Show that the reduction of [GPV08] is *history free*

(Classical) History-free Reduction

Classical RO Techniques:

(Classical) History-free Reduction

Classical RO Techniques:

- Simulating the random oracle.

(Classical) History-free Reduction

Classical RO Techniques:

- Simulating the random oracle.
    - Use a random oracle.

(Classical) History-free Reduction

Classical RO Techniques:

- Simulating the random oracle.
    - Use a random oracle.
- Determine what points the adversary is querying the oracle on.

(Classical) History-free Reduction

Classical RO Techniques:

- Simulating the random oracle.
    - Use a random oracle.
- Determine what points the adversary is querying the oracle on.
    - Not allowed.

(Classical) History-free Reduction

Classical RO Techniques:

- Simulating the random oracle.
  - Use a random oracle.
- Determine what points the adversary is querying the oracle on.
  - Not allowed.
- Programming the random oracle.

(Classical) History-free Reduction

Classical RO Techniques:

- Simulating the random oracle.
  - Use a random oracle.
- Determine what points the adversary is querying the oracle on.
  - Not allowed.
- Programming the random oracle.
  - Only non-adaptively (i.e. no knowledge of previous queries)

(Classical) History-free Reduction

Classical RO Techniques:

- Simulating the random oracle.
  - Use a random oracle.
- Determine what points the adversary is querying the oracle on.
  - Not allowed.
- Programming the random oracle.
  - Only non-adaptively (i.e. no knowledge of previous queries)
- Rewinding

(Classical) History-free Reduction

Classical RO Techniques:

- Simulating the random oracle.
  - Use a random oracle.
- Determine what points the adversary is querying the oracle on.
  - Not allowed.
- Programming the random oracle.
  - Only non-adaptively (i.e. no knowledge of previous queries)
- Rewinding
  - Not allowed.

(Classical) History-free Reduction

(Classical) History-free Reduction

- Reduction algorithm has private random oracle O_c
  - Implemented on the fly

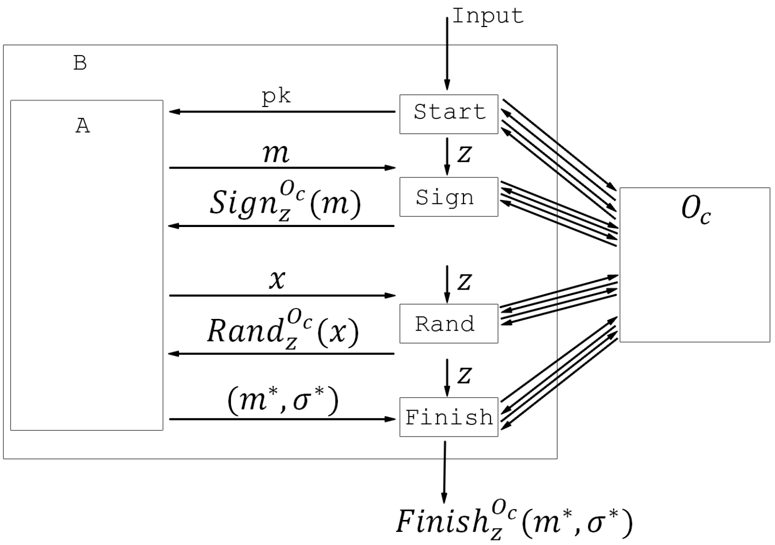
(Classical) History-free Reduction

- Reduction algorithm has private random oracle O_c
  - Implemented on the fly
- Random oracle queries answered by $Rand^{O_c}$
  - Truly random

(Classical) History-free Reduction

- Reduction algorithm has private random oracle O_c
    - Implemented on the fly
- Random oracle queries answered by $Rand^{O_c}$
    - Truly random
- Signatures answered by $Sign^{O_c}$
    - Consistent with random oracle
    - Distribution identical to actual

(Classical) History Free Reduction

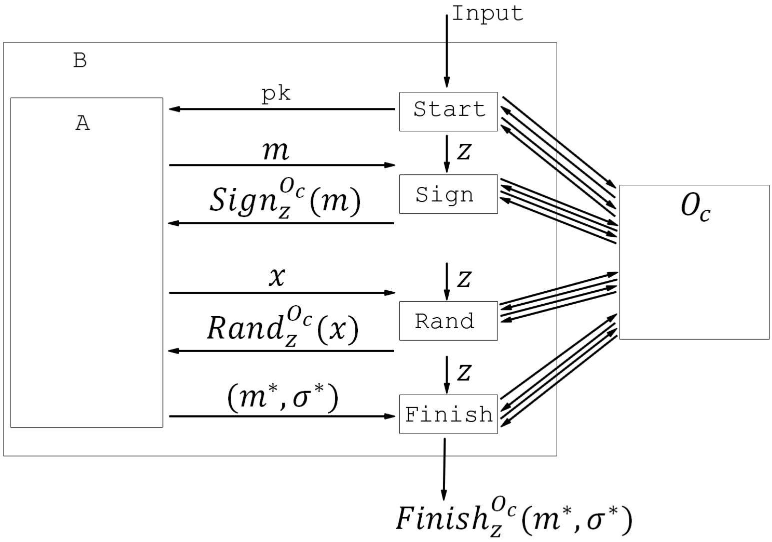


Main Theorem

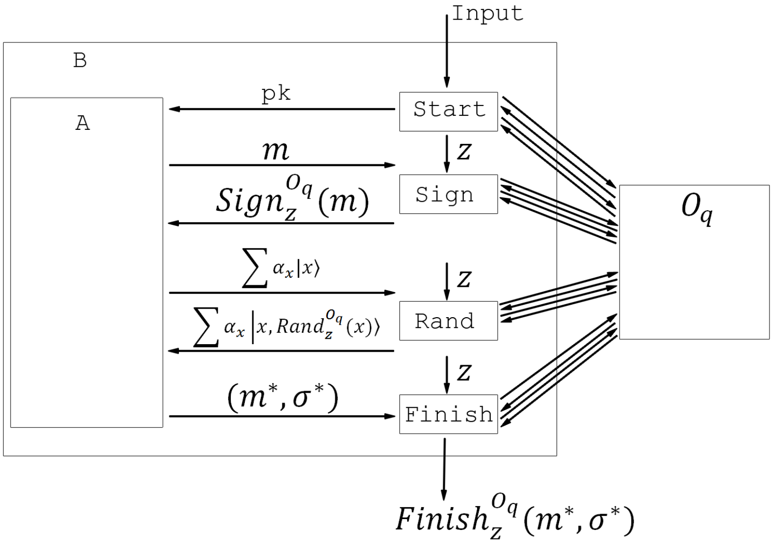
Theorem

Suppose a random oracle model signature scheme $\mathcal{S}$ has a history-free reduction that transforms any classical adversary A into a classical algorithm B for some hard problem for quantum computers. Suppose further that quantum pseudorandom functions exist. Then $\mathcal{S}$ is secure against quantum adversaries.

Proof


Input
B
A
pk
Start
m
z
Sign
$Sign_z^{O_c}(m)$
O_c
x
z
Rand
$Rand_z^{O_c}(x)$
z
(m^*, σ^*)
Finish
$Finish_z^{O_c}(m^*, \sigma^*)$

Proof


Input
B
A
pk
Start
m
z
$Sign_z^{O_q}(m)$
Sign
O_q
$\sum \alpha_x |x\rangle$
z
$\sum \alpha_x |x, Rand_z^{O_q}(x)\rangle$
Rand
z
(m^*, σ^*)
Finish
$Finish_z^{O_q}(m^*, \sigma^*)$

Problem

Quantum adversary could query on a superposition of exponentially many inputs.

Problem

Quantum adversary could query on a superposition of exponentially many inputs.

- Results in queries to O_q on exponential superposition.

Problem

Quantum adversary could query on a superposition of exponentially many inputs.

- Results in queries to O_q on exponential superposition.
- Implementing the random oracle would require exponential randomness.

Problem

Quantum adversary could query on a superposition of exponentially many inputs.

- Results in queries to O_q on exponential superposition.
- Implementing the random oracle would require exponential randomness.

Idea: Use a quantum pseudorandom function

Quantum PRF

A quantum pseudorandom function *PRF* is a keyed function that quantum computers cannot tell from a random oracle. Precisely, for all polynomial-time quantum oracle algorithms A,

$$\left|\Pr[A^{PRF_k}() = 1] - \Pr[A^{O_q}() = 1]\right| < \text{negl}$$

Where the left probability is over k and the right is over O_q, both chosen randomly.

Quantum PRF

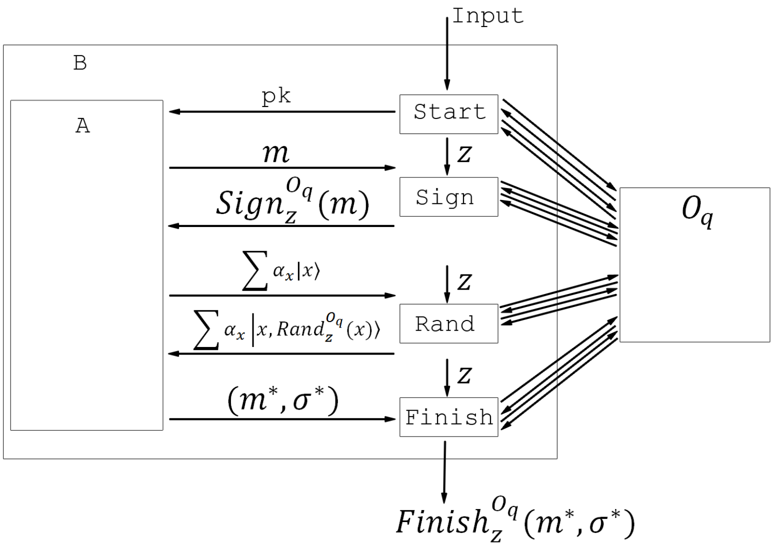
A quantum pseudorandom function PRF is a keyed function that quantum computers cannot tell from a random oracle. Precisely, for all polynomial-time quantum oracle algorithms A,

$$\left|\Pr[A^{PRF_k}() = 1] - \Pr[A^{O_q}() = 1]\right| < \mathrm{negl}$$

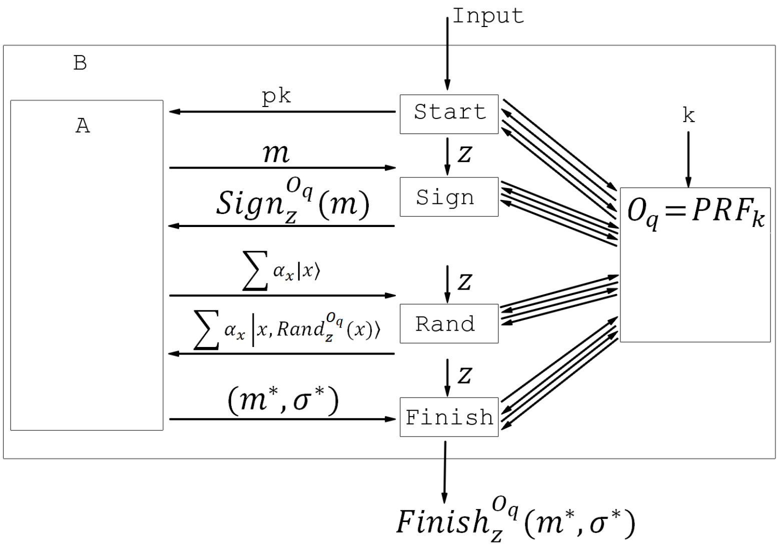
Where the left probability is over k and the right is over O_q, both chosen randomly.

No known provably secure constructions!

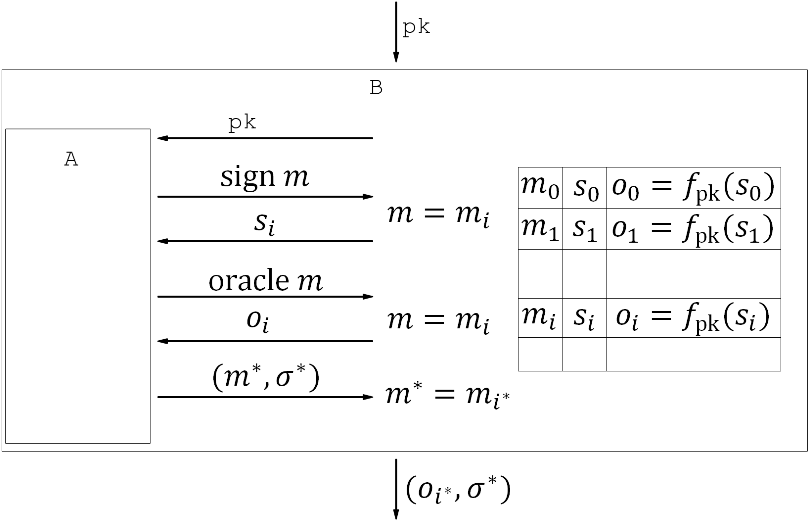
Proof


Input
B
A
pk
Start
m
z
Sign
$Sign_z^{O_q}(m)$
O_q
$\sum \alpha_x |x\rangle$
z
Rand
$\sum \alpha_x \left|x, Rand_z^{O_q}(x)\right\rangle$
z
(m^*, σ^*)
Finish
$Finish_z^{O_q}(m^*, \sigma^*)$

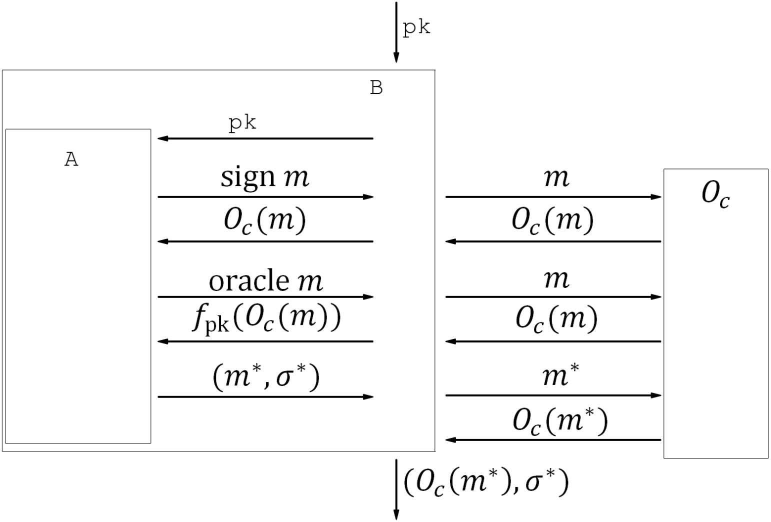
Proof


Input
B
A
pk
Start
m
z
$Sign_z^{O_q}(m)$
Sign
k
$O_q = PRF_k$
$\sum \alpha_x |x\rangle$
z
$\sum \alpha_x \left|x, Rand_z^{O_q}(x)\right\rangle$
Rand
z
(m^*, σ^*)
Finish
$Finish_z^{O_q}(m^*, \sigma^*)$

GPV Reduction



Modified GPV Reduction



History-Freeness of GPV Reduction

This reduction is in history-free form!

History-Freeness of GPV Reduction

This reduction is in history-free form!

Caveats:

- $f_{\mathrm{pk}}(r)$ for random r is NOT truly random for GPV construction.
- GPV signatures are NOT truly random preimages of $O(m)$

History-Freeness of GPV Reduction

This reduction is in history-free form!

Caveats:

- $f_{\mathrm{pk}}(r)$ for random r is NOT truly random for GPV construction.
- GPV signatures are NOT truly random preimages of $O(m)$
- Need to relax definition of history freeness to allow indistinguishable (by quantum adversaries)

Other History-Free Reductions

- Full Domain Hash from claw-free permutations ([Cor00]).
- Katz-Wang Signatures (KW03)

Encryption

Encryption

- History-freeness complicated by the challenge query. Easier to prove directly.

Encryption

- History-freeness complicated by the challenge query. Easier to prove directly.
- CPA-security of Bellare-Rogaway encryption scheme ([BR93]):

$$E_{\mathrm{pk}}(m) = f_{\mathrm{pk}}(r) || m \oplus O(r) \text{ for a random } r$$

where f is a trapdoor permutation.

Encryption

- History-freeness complicated by the challenge query. Easier to prove directly.
- CPA-security of Bellare-Rogaway encryption scheme ([BR93]):

$$E_{\mathrm{pk}}(m) = f_{\mathrm{pk}}(r) || m \oplus O(r) \text{ for a random } r$$

where f is a trapdoor permutation.

- CCA-security of hybrid encryption scheme:

$$E_{\mathrm{pk}}(m) = f_{\mathrm{pk}}(r) || \left(E_S\right)_{O(r)}(m) \text{ for a random r}$$

where f is a trapdoor permutation and E_S is CCA-secure private key encryption.

Conclusion

Conclusion

- Classical security reductions do not carry over to the quantum world

Conclusion

- Classical security reductions do not carry over to the quantum world
- Restricted class of classical security proofs do imply quantum security

Conclusion

- Classical security reductions do not carry over to the quantum world
- Restricted class of classical security proofs do imply quantum security
- GPV Signatures are secure

Open Problems

- Generic Full Domain Hash

Open Problems

- Generic Full Domain Hash
- Signatures from Identification Protocols [FS86]

Open Problems

- Generic Full Domain Hash
- Signatures from Identification Protocols [FS86]
- CCA-security from weaker security notions [FO99]

Open Problems

- Generic Full Domain Hash
- Signatures from Identification Protocols [FS86]
- CCA-security from weaker security notions [FO99]
- Quantum PRFs from one-way functions