RUHR-UNIVERSITÄT BOCHUM

Optimal Security Proofs for Full-Domain Hash, revisited

Cambridge, EUROCRYPT 2012

Saqib A. Kakvi Eike Kiltz
Foundations of Cryptography
Chair for Cryptography and IT Security

Horst Görtz Institut
für IT-Sicherheit

1 Introduction

2 Our results

3 Extensions

4 Conclusions

RSA-Full Domain Hash Signatures

- RSA-Full Domain Hash (RSA-FDH) was introduced by Bellare and Rogaway [BelRog93] and is arguably one of the most important signature schemes based on RSA.

RSA-Full Domain Hash Signatures

§ RSA-Full Domain Hash (RSA-FDH) was introduced by Bellare and Rogaway [BelRog93] and is arguably one of the most important signature schemes based on RSA.

procedure KeyGen $p, q \in_R \mathbb{P}$, $N = pq$ $e \in_R \mathbb{Z}_{\varphi(N)}$ Pick $H : \{0,1\}^* \to \mathbb{Z}_N$ return $(pk = (N, e, H), sk = (p, q))$	**procedure Sign**(sk, m) return $\sigma = H(m)^{\frac{1}{e}} \mod N$ **procedure Verify**(pk, m, σ) if $\sigma^e \mod N = H(m)$ then return 1 else return 0

RSA-Full Domain Hash Signatures

- RSA-Full Domain Hash (RSA-FDH) was introduced by Bellare and Rogaway [BelRog93] and is arguably one of the most important signature schemes based on RSA.

procedure KeyGen $p, q \in_R \mathbb{P},\ N = pq$ $e \in_R \mathbb{Z}_{\varphi(N)}$ Pick $H : \{0,1\}^* \rightarrow \mathbb{Z}_N$ return $(pk = (N, e, H), sk = (p, q))$	**procedure Sign**(sk, m) return $\sigma = H(m)^{\frac{1}{e}} \mod N$ **procedure Verify**(pk, m, σ) if $\sigma^e \mod N = H(m)$ then return 1 else return 0

- RSA-FDH signatures are unique.

Classical security results of RSA-FDH

- We would like a tight security proof (UF-CMA) for RSA-FDH.

Classical security results of RSA-FDH

- We would like a tight security proof (UF-CMA) for RSA-FDH.
- All known proofs are non-tight

Classical security results of RSA-FDH

- We would like a tight security proof (UF-CMA) for RSA-FDH.
- All known proofs are non-tight
- In practice, people use a 1024-bit modulus, but in theory?

Classical security results of RSA-FDH

- We would like a tight security proof (UF-CMA) for RSA-FDH.
- All known proofs are non-tight
- In practice, people use a 1024-bit modulus, but in theory?
- If RSA is (t, ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure

Security Proof	Security Loss ε/ε'	Equivalent RSA modulus

Classical security results of RSA-FDH

- We would like a tight security proof (UF-CMA) for RSA-FDH.
- All known proofs are non-tight
- In practice, people use a 1024-bit modulus, but in theory?
- If RSA is (t, ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure

Security Proof	Security Loss ε/ε'	Equivalent RSA modulus
Ideal	1	$\approx$ 1024 bits

Classical security results of RSA-FDH

- We would like a tight security proof (UF-CMA) for RSA-FDH.
- All known proofs are non-tight
- In practice, people use a 1024-bit modulus, but in theory?
- If RSA is (t, ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure

Security Proof	Security Loss ε/ε'	Equivalent RSA modulus
~~Ideal~~	~~1~~	~~1024 bits~~

Classical security results of RSA-FDH

- We would like a tight security proof (UF-CMA) for RSA-FDH.
- All known proofs are non-tight
- In practice, people use a 1024-bit modulus, but in theory?
- If RSA is (t, ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure

Security Proof	Security Loss ε/ε'	Equivalent RSA modulus
~~Ideal~~	~~1~~	~~1024 bits~~
[BelRog93]	$q_h \approx 2^{60}$	≈ 200 bits

Classical security results of RSA-FDH

- We would like a tight security proof (UF-CMA) for RSA-FDH.
- All known proofs are non-tight
- In practice, people use a 1024-bit modulus, but in theory?
- If RSA is (t, ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure

Security Proof	Security Loss ε/ε'	Equivalent RSA modulus
~~Ideal~~	~~1~~	~~1024 bits~~
[BelRog93]	$q_h \approx 2^{60}$	$\approx$ 200 bits
[Coron00]	$q_s \approx 2^{30}$	$\approx$ 500 bits

Classical security results of RSA-FDH

- We would like a tight security proof (UF-CMA) for RSA-FDH.
- All known proofs are non-tight
- In practice, people use a 1024-bit modulus, but in theory?
- If RSA is (t, ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure

Security Proof	Security Loss ε/ε'	Equivalent RSA modulus
~~Ideal~~	~~1~~	~~1024 bits~~
[BelRog93]	$q_h \approx 2^{60}$	$\approx$ 200 bits
[Coron00]	$q_s \approx 2^{30}$	$\approx$ 500 bits
(PSS) [BelRog96]	1	$\approx$ 1024 bits

Classical security results of RSA-FDH

- We would like a tight security proof (UF-CMA) for RSA-FDH.
- All known proofs are non-tight
- In practice, people use a 1024-bit modulus, but in theory?
- If RSA is (t, ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure

Security Proof	Security Loss ε/ε'	Equivalent RSA modulus
~~Ideal~~	~~1~~	~~1024 bits~~
[BelRog93]	$q_h \approx 2^{60}$	$\approx$ 200 bits
[Coron00]	$q_s \approx 2^{30}$	$\approx$ 500 bits
(PSS) [BelRog96]	1	$\approx$ 1024 bits

- Can RSA-FDH be tightly secure?

Classical security results of RSA-FDH

- We would like a tight security proof (UF-CMA) for RSA-FDH.
- All known proofs are non-tight
- In practice, people use a 1024-bit modulus, but in theory?
- If RSA is (t, ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure

Security Proof	Security Loss ε/ε'	Equivalent RSA modulus
~~Ideal~~	~~1~~	~~1024 bits~~
[BelRog93]	$q_h \approx 2^{60}$	$\approx$ 200 bits
[Coron00]	$q_s \approx 2^{30}$	$\approx$ 500 bits
(PSS) [BelRog96]	1	$\approx$ 1024 bits

- Can RSA-FDH be tightly secure?
- Exactly 10 years ago at EUROCRYPT 2002 in Amsterdam, Coron answered this by showing that a loss of a factor of q_s is optimal

1 Introduction

2 Our results

3 Extensions

4 Conclusions

Our Results

- We revisit Coron's impossibility result

Our Results

- We revisit Coron's impossibility result
  - Uncover a subtle flaw

Our Results

- We revisit Coron's impossibility result
  - Uncover a subtle flaw
  - Proof does not hold for small e

Our Results

- We revisit Coron's impossibility result
  - Uncover a subtle flaw
  - Proof does not hold for small e
- We show a tight proof for small e

Our Results

- We revisit Coron's impossibility result
  - Uncover a subtle flaw
  - Proof does not hold for small e
- We show a tight proof for small e
  - Proof is to Φ-Hiding , which is stronger than RSA

Our Results

- We revisit Coron's impossibility result
  - Uncover a subtle flaw
  - Proof does not hold for small e
- We show a tight proof for small e
  - Proof is to Φ-Hiding , which is stronger than RSA
- We then show some generalizations and extensions.

Flaw in Coron's Proof

Theorem 1 (Coron)

If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.

Flaw in Coron's Proof

Theorem 1 (Coron)

If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.

$\mathcal{R}$

Flaw in Coron's Proof

Theorem 1 (Coron)

If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.

$$(N, e, x^e \mod N)$$

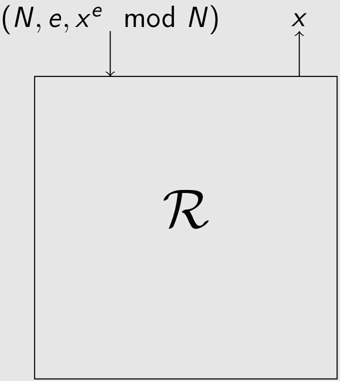
$$\downarrow$$

$$\boxed{\mathcal{R}}$$

Flaw in Coron's Proof

Theorem 1 (Coron)

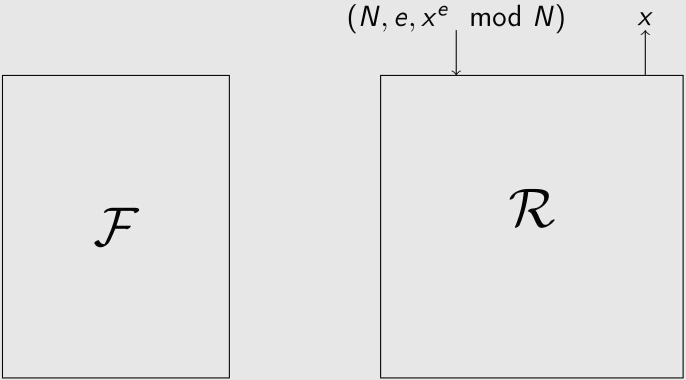
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Flaw in Coron's Proof

Theorem 1 (Coron)

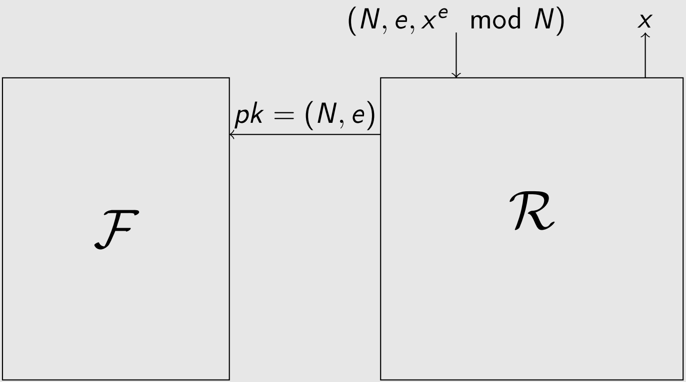
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Flaw in Coron's Proof

Theorem 1 (Coron)

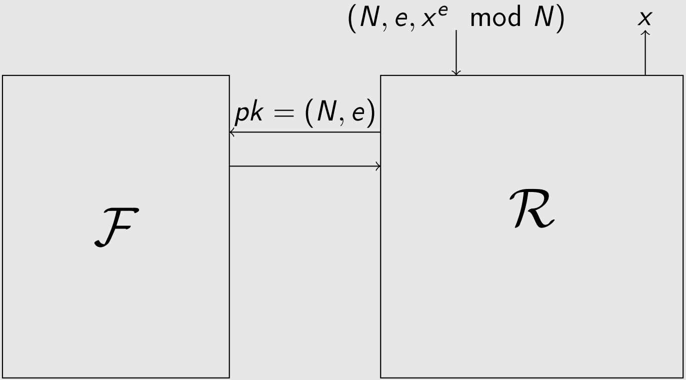
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Flaw in Coron's Proof

Theorem 1 (Coron)

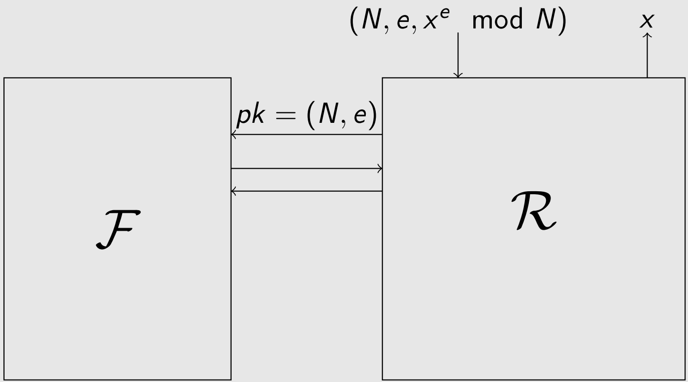
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Flaw in Coron's Proof

Theorem 1 (Coron)

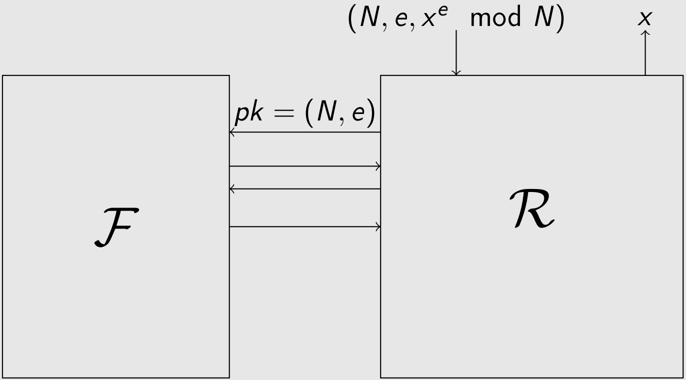
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Flaw in Coron's Proof

Theorem 1 (Coron)

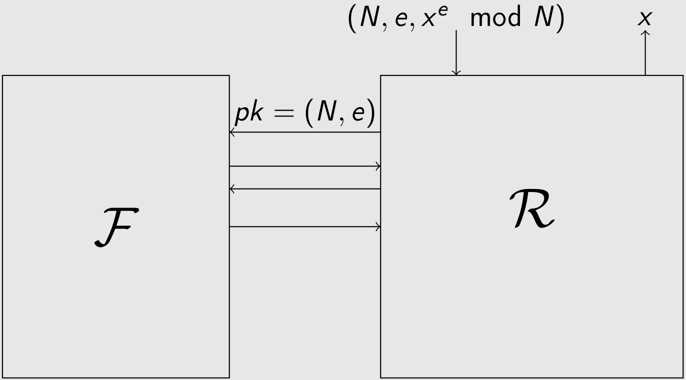
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Flaw in Coron’s Proof

Theorem 1 (Coron)

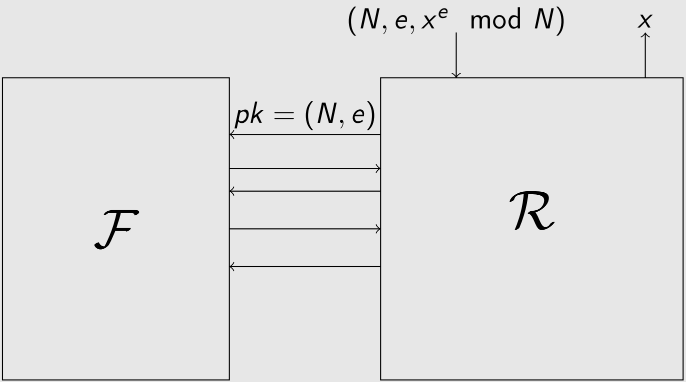
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Flaw in Coron's Proof

Theorem 1 (Coron)

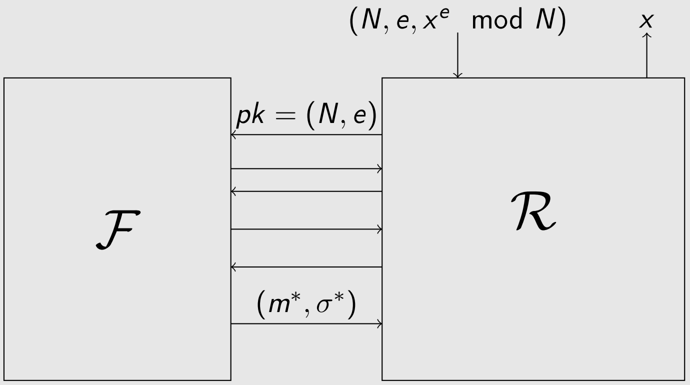
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Flaw in Coron's Proof

Theorem 1 (Coron)

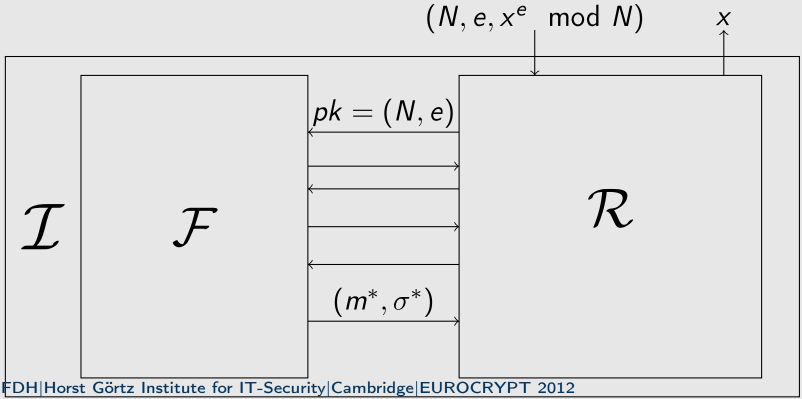
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Flaw in Coron's Proof

Theorem 1 (Coron)

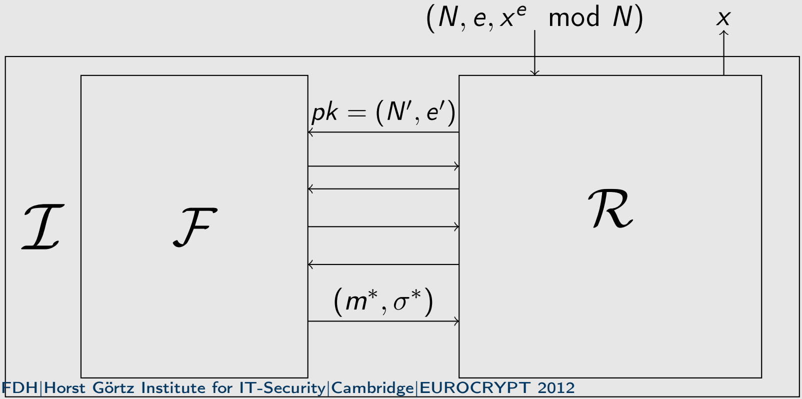
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Fixing Coron's Proof

Theorem 2 (Coron Corrected)

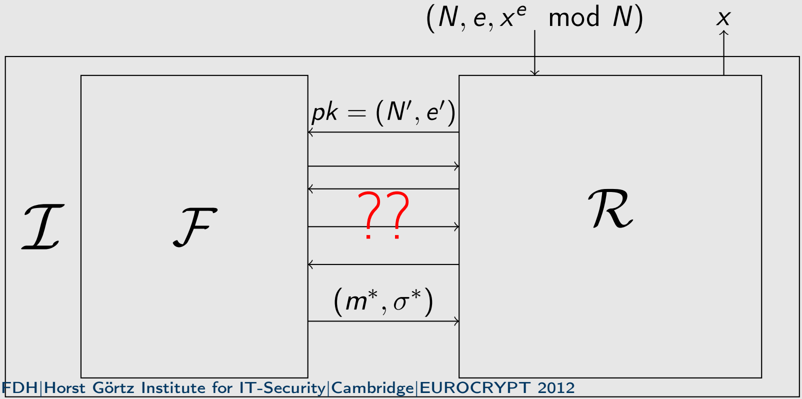
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Fixing Coron's Proof

Theorem 2 (Coron Corrected)

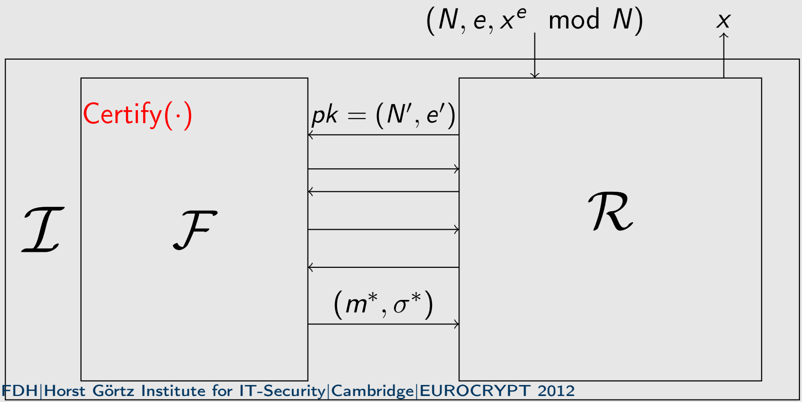
If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting RSA, with security loss less than q_s, then we can efficiently invert RSA.



Fixing Coron's Proof

Theorem 2 (Coron Corrected)

If there is a reduction $\mathcal{R}$ from RSA-FDH to inverting certified RSA, with security loss less than q_s, then we can efficiently invert RSA.



Is RSA certified?

- We say RSA is certified if given a public key (N, e), we can decide in polynomial time if the RSA function $f_{(N,e)}(x) = x^e \mod N$ is a permutation. [BelYun03]

Is RSA certified?

- We say RSA is certified if given a public key (N, e), we can decide in polynomial time if the RSA function $f_{(N,e)}(x) = x^e \mod N$ is a permutation. [BelYun03]
- Need to decide if $e|\varphi(N)$ or if $gcd(e, \varphi(N)) = 1$.

Is RSA certified?

- We say RSA is certified if given a public key (N, e), we can decide in polynomial time if the RSA function $f_{(N,e)}(x) = x^e \mod N$ is a permutation. [BelYun03]
- Need to decide if $e|\varphi(N)$ or if $gcd(e, \varphi(N)) = 1$.
- This is easy for prime $e > N$.

Is RSA certified?

- We say RSA is certified if given a public key (N, e), we can decide in polynomial time if the RSA function $f_{(N,e)}(x) = x^e \mod N$ is a permutation. [BelYun03]
- Need to decide if $e|\varphi(N)$ or if $gcd(e, \varphi(N)) = 1$.
- This is easy for prime $e > N$.
- Thought to be hard for prime $e < N^{0.25}$.

Is RSA certified?

- We say RSA is certified if given a public key (N, e), we can decide in polynomial time if the RSA function $f_{(N,e)}(x) = x^e \mod N$ is a permutation. [BelYun03]
- Need to decide if $e|\varphi(N)$ or if $gcd(e, \varphi(N)) = 1$.
- This is easy for prime $e > N$.
- Thought to be hard for prime $e < N^{0.25}$.
- Overall, we have:

Figure: Known results for RSA Certification with Prime Exponent

Is RSA certified?

- We say RSA is certified if given a public key (N, e), we can decide in polynomial time if the RSA function $f_{(N,e)}(x) = x^e \mod N$ is a permutation. [BelYun03]
- Need to decide if $e|\varphi(N)$ or if $gcd(e, \varphi(N)) = 1$.
- This is easy for prime $e > N$.
- Thought to be hard for prime $e < N^{0.25}$.
- Overall, we have:

Figure: Known results for RSA Certification with Prime Exponent

Is RSA certified?

- We say RSA is certified if given a public key (N, e), we can decide in polynomial time if the RSA function $f_{(N,e)}(x) = x^e \mod N$ is a permutation. [BelYun03]
- Need to decide if $e | \varphi(N)$ or if $gcd(e, \varphi(N)) = 1$.
- This is easy for prime $e > N$.
- Thought to be hard for prime $e < N^{0.25}$.
- Overall, we have:

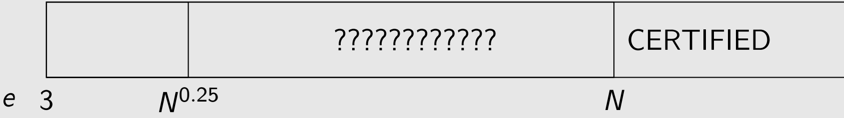


Figure: Known results for RSA Certification with Prime Exponent

Is RSA certified?

- We say RSA is certified if given a public key (N, e), we can decide in polynomial time if the RSA function $f_{(N,e)}(x) = x^e \mod N$ is a permutation. [BelYun03]
- Need to decide if $e | \varphi(N)$ or if $gcd(e, \varphi(N)) = 1$.
- This is easy for prime $e > N$.
- Thought to be hard for prime $e < N^{0.25}$.
- Overall, we have:

LOSSY	????????????	CERTIFIED

e 3 $N^{0.25}$ N

Figure: Known results for RSA Certification with Prime Exponent

Is RSA certified?

- We say RSA is certified if given a public key (N, e), we can decide in polynomial time if the RSA function $f_{(N,e)}(x) = x^e \mod N$ is a permutation. [BelYun03]
- Need to decide if $e | \varphi(N)$ or if $gcd(e, \varphi(N)) = 1$.
- This is easy for prime $e > N$.
- Thought to be hard for prime $e < N^{0.25}$.
- Overall, we have:

Figure: Known results for RSA Certification with Prime Exponent

Lossiness of RSA

- A function is said to be lossy if there exists an alternate KeyGen algorithm that outputs a lossy key [PeiWat08].

Lossiness of RSA

- A function is said to be lossy if there exists an alternate KeyGen algorithm that outputs a lossy key [PeiWat08].
- A lossy keys are computationally indistinguishable real keys

Lossiness of RSA

- A function is said to be lossy if there exists an alternate KeyGen algorithm that outputs a lossy key [PeiWat08].
- A lossy keys are computationally indistinguishable real keys
- Lossy keys give a function where the range is smaller than the domain.

Lossiness of RSA

§ A function is said to be lossy if there exists an alternate KeyGen algorithm that outputs a lossy key [PeiWat08].

§ A lossy keys are computationally indistinguishable real keys

§ Lossy keys give a function where the range is smaller than the domain.

§ In particular for RSA, the lossy function is e-to-1.

Lossines of RSA

- RSA was shown to be lossy under Φ-Hiding [KOS10].

Lossines of RSA

§ RSA was shown to be lossy under Φ-Hiding [KOS10].

§ Φ-Hiding was introduced in 1999 by Cachin, Micali and Stadler [CMS99].

Lossines of RSA

- RSA was shown to be lossy under Φ-Hiding [KOS10].
- Φ-Hiding was introduced in 1999 by Cachin, Micali and Stadler [CMS99].
- Φ-Hiding states that given N and a prime $e < N^{0.25}$ it is hard to distinguish $e|\varphi(N)$ and $gcd(e, \varphi(N)) = 1$.

Main Result

Main Theorem

If Φ-Hiding is (t', ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure, for any q_h, q_s, with $t \approx t', \varepsilon \approx 2\varepsilon'$.

Main Result

Main Theorem

If Φ-Hiding is (t', ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure, for any q_h, q_s, with $t \approx t', \varepsilon \approx 2\varepsilon'$.

- GAME0 Standard UF-CMA

Main Result

Main Theorem

If Φ-Hiding is (t', ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure, for any q_h, q_s, with $t \approx t', \varepsilon \approx 2\varepsilon'$.

- GAME0 Standard UF-CMA
- GAME1 Simulate H such that sign no longer needs sk. Simulation knows exactly 1 valid signature for each message

Main Result

Main Theorem

If Φ-Hiding is (t', ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure, for any q_h, q_s, with $t \approx t', \varepsilon \approx 2\varepsilon'$.

- GAME0 Standard UF-CMA
- GAME1 Simulate H such that sign no longer needs sk. Simulation knows exactly 1 valid signature for each message
- GAME2 KeyGen is switched from real to lossy. Now each message has exactly e valid signatures.

Main Result

Main Theorem

If Φ-Hiding is (t', ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure, for any q_h, q_s, with $t \approx t', \varepsilon \approx 2\varepsilon'$.

- GAME0 Standard UF-CMA
- GAME1 Simulate H such that sign no longer needs sk. Simulation knows exactly 1 valid signature for each message
- GAME2 KeyGen is switched from real to lossy. Now each message has exactly e valid signatures.
- A forgery (m^*, σ^*) gives a collision in the RSA function with probability $1 - \frac{1}{e}$, allowing us to factor or break Φ-Hiding .

Main Result

Main Theorem

If Φ-Hiding is (t', ε')-hard, then RSA-FDH is $(q_h, q_s, t, \varepsilon)$-secure, for any q_h, q_s, with $t \approx t', \varepsilon \approx 2\varepsilon'$.

- GAME0 Standard UF-CMA
- GAME1 Simulate H such that sign no longer needs sk. Simulation knows exactly 1 valid signature for each message
- GAME2 KeyGen is switched from real to lossy. Now each message has exactly e valid signatures.
- A forgery (m^*, σ^*) gives a collision in the RSA function with probability $1 - \frac{1}{e}$, allowing us to factor or break Φ-Hiding .
- The final security loss is approximately $2 = O(1)$.

Implications of our results

- If we assume that solving Φ-Hiding is equivalent to inverting RSA, then:

Security Proof	Security Loss ε/ε'	Equivalent RSA modulus
~~Ideal~~	~~1~~	~~1024 bits~~
[BelRog93]	$q_h \approx 2^{60}$	$\approx$ 200 bits
[Coron00]	$q_s \approx 2^{30}$	$\approx$ 500 bits

Implications of our results

- If we assume that solving Φ-Hiding is equivalent to inverting RSA, then:

Security Proof	Security Loss ε/ε'	Equivalent RSA modulus
This work	1	$\approx$ 1024 bits
[BelRog93]	$q_h \approx 2^{60}$	$\approx$ 200 bits
[Coron00]	$q_s \approx 2^{30}$	$\approx$ 500 bits

1 Introduction

2 Our results

3 Extensions

4 Conclusions

Extensions: Generalizations

- We can extend our main theorem to any certified trapdoor permutation

Theorem 3

If TDP is (t', ε')-lossy, then TDP-FDH is $(q_h, q_s, t, \varepsilon)$-secure, for any q_h, q_s, with $t \approx t', \varepsilon \approx 2\varepsilon'$.

Extensions: Generalizations

- We can extend our main theorem to any certified trapdoor permutation

Theorem 3

If TDP is (t', ε')-lossy, then TDP-FDH is $(q_h, q_s, t, \varepsilon)$-secure, for any q_h, q_s, with $t \approx t', \varepsilon \approx 2\varepsilon'$.

- We can show impossibility for any hard problem Π and any certified unique signature scheme Σ.

Theorem 4

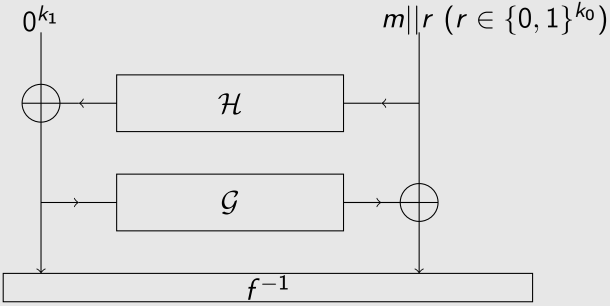
If there is a reduction $\mathcal{R}$ from Σ to solving Π, with security loss less than q_s, then we can efficiently solve Π.

Extensions: PSS

- Our results also extend to PSS, in particular PSS-R.

Extensions: PSS

- Our results also extend to PSS, in particular PSS-R.

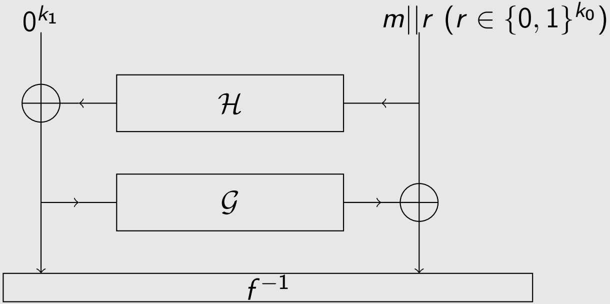


Extensions: PSS

- Our results also extend to PSS, in particular PSS-R.

Theorem 5

If Φ-Hiding is (t', ε')-hard, then RSA-PSS-R is $(q_h, q_s, t, \varepsilon)$-secure, for any q_h, q_s, with $t \approx t', \varepsilon \approx 2 \cdot \varepsilon' + \frac{(q_h+q_s)^2}{2^{k_1}}$



Extensions: PSS with message recovery

- When using signatures with recovery we want to minimize bandwidth.

Extensions: PSS with message recovery

- When using signatures with recovery we want to minimize bandwidth.
- Signer needs only to send the "enhanced signature".

Extensions: PSS with message recovery

- When using signatures with recovery we want to minimize bandwidth.
- Signer needs only to send the "enhanced signature".
- Verify is replaced by Recover, which outputs message or $\perp$.

Extensions: PSS with message recovery

- When using signatures with recovery we want to minimize bandwidth.
- Signer needs only to send the "enhanced signature".
- Verify is replaced by Recover, which outputs message or $\perp$.
- We use a measure called overhead, which is the difference in size between the message and the "enhanced signature".

Extensions: PSS with message recovery

- When using signatures with recovery we want to minimize bandwidth.
- Signer needs only to send the "enhanced signature".
- Verify is replaced by Recover, which outputs message or $\perp$.
- We use a measure called overhead, which is the difference in size between the message and the "enhanced signature".

Extensions: PSS with message recovery

- When using signatures with recovery we want to minimize bandwidth.
- Signer needs only to send the "enhanced signature".
- Verify is replaced by Recover, which outputs message or $\perp$.
- We use a measure called overhead, which is the difference in size between the message and the "enhanced signature".

Security Proof	Randomness	Padding	Total overhead
Bellare-Rogaway [BR96]	160	160	320
Coron [Cor02]	30	160	190
This work	0	160	160

Table: Total overhead using RSA-PSS-R for 80 bit security.

Extensions: PSS with message recovery

- When using signatures with recovery we want to minimize bandwidth.
- Signer needs only to send the "enhanced signature".
- Verify is replaced by Recover, which outputs message or $\perp$.
- We use a measure called overhead, which is the difference in size between the message and the "enhanced signature".

Security Proof	Randomness	Padding	Total overhead
Bellare-Rogaway [BR96]	160	160	320
Coron [Cor02]	30	160	190
This work	0	160	160

Table: Total overhead using RSA-PSS-R for 80 bit security.

- PSS-R comparable to BLS signatures.

1 Introduction

2 Our results

3 Extensions

4 Conclusions

Conclusion

- Revisited and corrected Coron's proof.

Conclusion

- Revisited and corrected Coron's proof.
- Tight security proof for RSA-FDH with small exponents.

Conclusion

- Revisited and corrected Coron's proof.
- Tight security proof for RSA-FDH with small exponents.
- Extensions to TDP and other problems.

Conclusion

- Revisited and corrected Coron's proof.
- Tight security proof for RSA-FDH with small exponents.
- Extensions to TDP and other problems.
- Extensions to PSS and PSS-R.

Many thanks for your attention!

QUESTIONS?