

# Approximate Divisor Multiples Factoring with Only a Third of the Secret CRT-Exponents

EUROCRYPT'22

---

Alexander May<sup>1</sup>   **Julian Nowakowski**<sup>1</sup>   Santanu Sarkar<sup>2</sup>

<sup>1</sup> Ruhr-University Bochum, Germany

<sup>2</sup> Indian Institute of Technology Madras, India

<https://eprint.iacr.org/2022/271>

RSA public key:

$$N = pq$$

$$e$$

RSA private key:

$$N = pq$$

$$d = e^{-1} \bmod (p-1)(q-1)$$

CRT-RSA public key:

$$N = pq$$

$$e$$

CRT-RSA private key:

$$N = pq$$

$$d = e^{-1} \bmod (p-1)(q-1)$$

$$p$$

$$q$$

$$d_p = d \bmod (p-1)$$

$$d_q = d \bmod (q-1)$$

$$q_{inv} = q^{-1} \bmod p$$

CRT-RSA public key:

$$N = pq$$

$$e$$

CRT-RSA private key:

~~$$N = pq$$~~

~~$$d = e^{-1} \bmod (p-1)(q-1)$$~~

$$p$$

~~$$q$$~~

~~$$d_p = d \bmod (p-1)$$~~

~~$$d_q = d \bmod (q-1)$$~~

~~$$q_{inv} = q^{-1} \bmod p$$~~

CRT-RSA public key:

$$N = pq$$

$$e$$

CRT-RSA private key:

~~$$N = pq$$~~

~~$$d = e^{-1} \bmod (p-1)(q-1)$$~~

~~$$p$$~~

~~$$q$$~~

~~$$d_p = d \bmod (p-1)$$~~

~~$$d_q = d \bmod (q-1)$$~~

~~$$q_{inv} = q^{-1} \bmod p$$~~

## Partial Key Exposure Attacks

### Theorem (Coppersmith EC'96)

Given half of the bits of  $p$ , we can factor  $N$  in polynomial time.

# Partial Key Exposure Attacks

## Theorem (Coppersmith EC'96)

Given half of the bits of  $p$ , we can factor  $N$  in polynomial time.

**Coppersmith's attack is efficient:**

Bit-size of $N$	Runtime on a laptop
1024	$\approx 2\text{min}$
2048	$\approx 6\text{min}$
4096	$\approx 24\text{min}$

# Partial Key Exposure Attacks

## Theorem (Coppersmith EC'96)

Given half of the bits of  $p$ , we can factor  $N$  in polynomial time.

**Coppersmith's attack is efficient:**

Bit-size of $N$	Runtime on a laptop
1024	$\approx 2\text{min}$
2048	$\approx 6\text{min}$
4096	$\approx 24\text{min}$

**Coppersmith's attack is practical:**

- [BCC+13] breaks  $\approx 80$  smart cards.



# Partial Key Exposure Attacks

## Theorem (Coppersmith EC'96)

Given half of the bits of  $p$ , we can factor  $N$  in polynomial time.

**Coppersmith's attack is efficient:**

Bit-size of $N$	Runtime on a laptop
1024	$\approx 2\text{min}$
2048	$\approx 6\text{min}$
4096	$\approx 24\text{min}$

**Coppersmith's attack is practical:**

- [BCC+13] breaks  $\approx 80$  smart cards.
- [NSS+17] breaks  $\approx 10^7$  smart cards.

# Partial Key Exposure Attacks

## Theorem (Coppersmith EC'96)

Given half of the bits of  $p$ , we can factor  $N$  in polynomial time.

**Coppersmith's attack is efficient:**

Bit-size of $N$	Runtime on a laptop
1024	$\approx 2\text{min}$
2048	$\approx 6\text{min}$
4096	$\approx 24\text{min}$

**Coppersmith's attack is practical:**

- [BCC+13] breaks  $\approx 80$  smart cards.
- [NSS+17] breaks  $\approx 10^7$  smart cards.

## Theorem (Boneh, Durfee, Frankel AC'98)

Suppose  $e = \mathcal{O}(\log N)$ . Given a quarter of the bits of  $d$ , we can factor  $N$  in polynomial time.

# Partial Key Exposure Attacks

## Theorem (Coppersmith EC'96)

Given half of the bits of  $p$ , we can factor  $N$  in polynomial time.

**Coppersmith's attack is efficient:**

Bit-size of $N$	Runtime on a laptop
1024	$\approx 2\text{min}$
2048	$\approx 6\text{min}$
4096	$\approx 24\text{min}$

**Coppersmith's attack is practical:**

- [BCC+13] breaks  $\approx 80$  smart cards.
- [NSS+17] breaks  $\approx 10^7$  smart cards.

## Theorem (Boneh, Durfee, Frankel AC'98)

Suppose  $e = \mathcal{O}(\log N)$ . Given a quarter of the bits of  $d$ , we can factor  $N$  in polynomial time.

## Theorem (Blömer, May CRYPTO'03)

Suppose  $e = \mathcal{O}(\log N)$ . Given half of the bits of  $d_p$ , we can factor  $N$  in polynomial time.

# Partial Key Exposure Attacks

## Theorem (Coppersmith EC'96)

Given half of the bits of  $p$ , we can factor  $N$  in polynomial time.

**Coppersmith's attack is efficient:**

Bit-size of $N$	Runtime on a laptop
1024	$\approx 2\text{min}$
2048	$\approx 6\text{min}$
4096	$\approx 24\text{min}$

**Coppersmith's attack is practical:**

- [BCC+13] breaks  $\approx 80$  smart cards.
- [NSS+17] breaks  $\approx 10^7$  smart cards.

## Theorem (Boneh, Durfee, Frankel AC'98)

Suppose  $e = \mathcal{O}(\log N)$ . Given a quarter of the bits of  $d$ , we can factor  $N$  in polynomial time.

## Theorem (Blömer, May CRYPTO'03)

Suppose  $e = \mathcal{O}(\log N)$ . Given half of the bits of  $d_p$ , we can factor  $N$  in polynomial time.

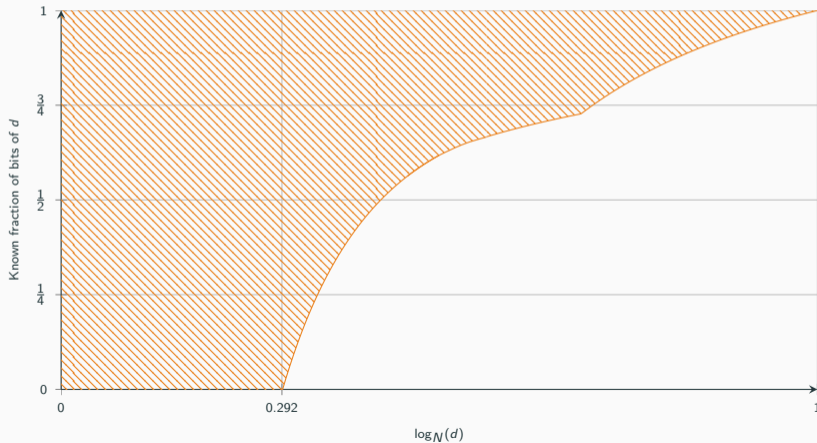
- For  $n$ -bit  $N$ , these attacks require  $\frac{n}{4}$  bits.

$$\mathbb{P} p \approx N^{1/2}, d \approx N, d_p \approx N^{1/2}.$$

# Partial Key Exposure Attacks

**Theorem (Ernst, Jochemsz, May, de Weger EC'05; Aono PKC'09; Takayasu, Kunihiro SAC'14)**

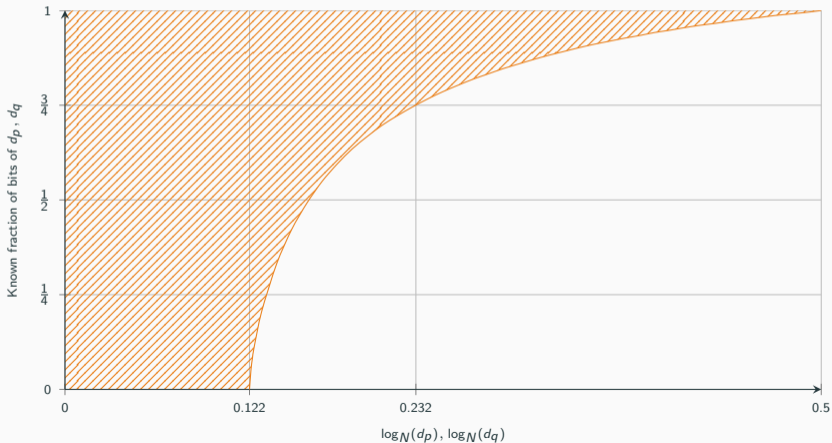
Suppose  $e = \mathcal{O}(N)$ . The smaller  $d$ , the less bits of  $d$  we have to know to factor  $N$  in polynomial time (assuming a well-established heuristic).



# Partial Key Exposure Attacks

## Theorem (May, N., Sarkar AC'21)

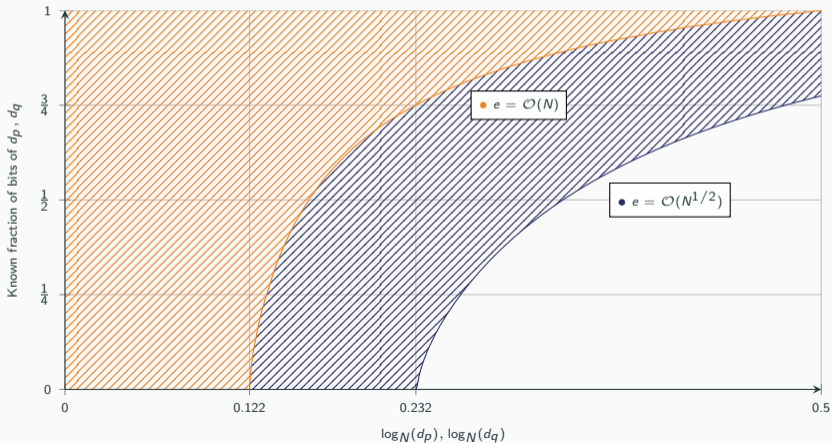
Suppose  $e = \mathcal{O}(N)$ . The smaller  $d_p, d_q$ , the less bits of  $d_p, d_q$  we have to know to factor  $N$  in polynomial time (assuming a well-established heuristic).



# Partial Key Exposure Attacks

## Theorem (May, N., Sarkar AC'21)

Suppose  $e = \mathcal{O}(N)$ . The smaller  $d_p, d_q$ , the less bits of  $d_p, d_q$  we have to know to factor  $N$  in polynomial time (assuming a well-established heuristic).



### Partial Key Exposure attacks in a nutshell:

- The smaller  $e$ ,  $d$ ,  $d_p$ ,  $d_q$ , the less bits we have to know to factor  $N$  in polynomial time.



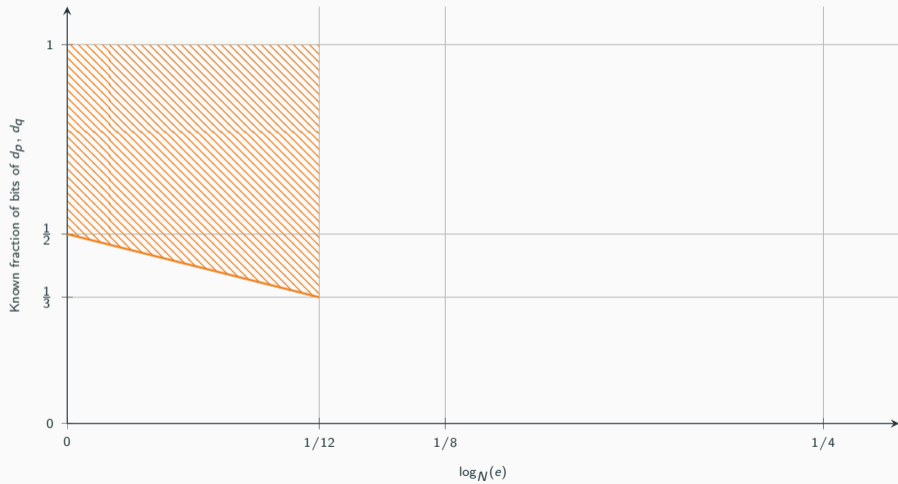
## Partial Key Exposure attacks in a nutshell:

- The smaller  $e$ ,  $d$ ,  $d_p$ ,  $d_q$ , the less bits we have to know to factor  $N$  in polynomial time.

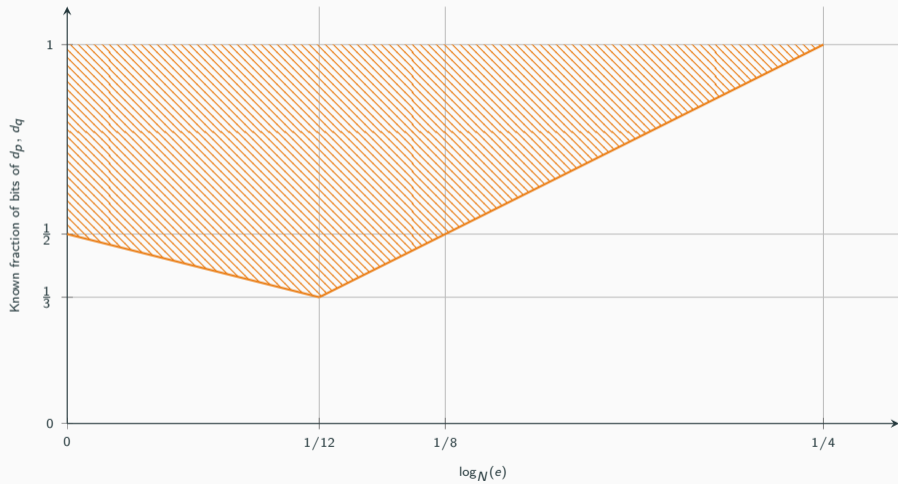
## Our result:

- New Partial Key Exposure attack for exposed  $d_p, d_q$  and small(-ish)  $e < N^{1/4}$ .
- **Surprising behaviour for  $e \leq N^{1/12}$ :**  
The larger  $e$ , the less bits we have to know to factor  $N$  in polynomial time.

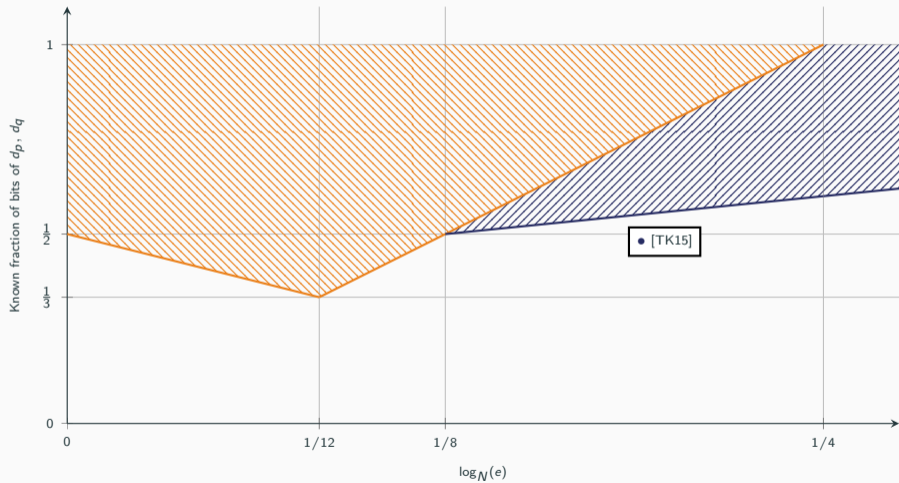
# Our Partial Key Exposure Attack



# Our Partial Key Exposure Attack



# Our Partial Key Exposure Attack



# Why Our Attack Behaves Differently

The usual strategy for RSA Partial Key Exposure attacks:

Model problem as system of polynomial equations

$$\begin{cases} f_1(x_1, \dots, x_k) = 0 \\ \vdots \\ f_n(x_1, \dots, x_k) = 0 \end{cases}.$$

Apply Coppersmith's method.

Our new strategy:

Model problem as system of polynomial equations

$$\begin{cases} f_1(x_1, \dots, x_k) = 0 \\ \vdots \\ f_n(x_1, \dots, x_k) = 0 \end{cases}.$$

Compute partial solution in few variables.

Apply Coppersmith's method.

## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$ed_p = 1 + k(p - 1),$$

$$ed_q = 1 + \ell(q - 1).$$

## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$ed_p = 1 + k(p - 1),$$

$$ed_q = 1 + \ell(q - 1).$$

## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$ed_p = 1 + k(p - 1),$$

$$ed_q = 1 + \ell(q - 1).$$



## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$ed_p = 1 + k(p - 1),$$

$$ed_q = 1 + \ell(q - 1).$$

### Question

How difficult is computing  $k, \ell$ ?

- Folklore: If  $e = \mathcal{O}(\log N)$ , then brute-force search runs in polynomial time.
- [GHM05]: If  $e \geq N^{1/4}$ , then as hard as factoring.

## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$ed_p = 1 + k(p - 1),$$

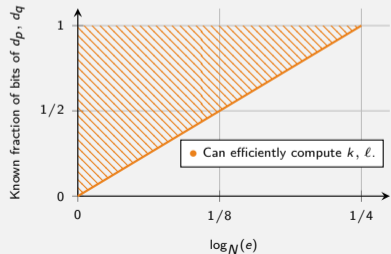
$$ed_q = 1 + \ell(q - 1).$$

### Question

How difficult is computing  $k, \ell$ ?

- Folklore: If  $e = \mathcal{O}(\log N)$ , then brute-force search runs in polynomial time.
- [GHM05]: If  $e \geq N^{1/4}$ , then as hard as factoring.

**Our result:**



## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$e(d_p^{\text{MSB}} \parallel d_p^{\text{LSB}}) = 1 + k(p - 1),$$

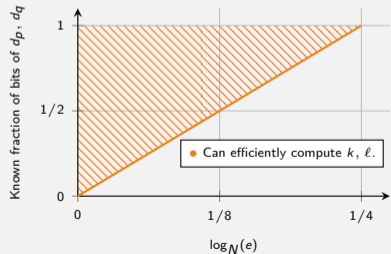
$$e(d_q^{\text{MSB}} \parallel d_q^{\text{LSB}}) = 1 + \ell(q - 1).$$

### Question

How difficult is computing  $k, \ell$ ?

- Folklore: If  $e = \mathcal{O}(\log N)$ , then brute-force search runs in polynomial time.
- [GHM05]: If  $e \geq N^{1/4}$ , then as hard as factoring.

**Our result:**



## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$e(d_p^{\text{MSB}} \| d_p^{\text{LSB}}) = 1 + k(p - 1) \approx kp,$$

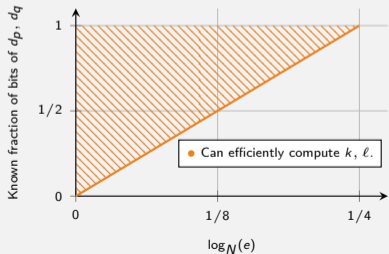
$$e(d_q^{\text{MSB}} \| d_q^{\text{LSB}}) = 1 + \ell(q - 1) \approx \ell q.$$

### Question

How difficult is computing  $k, \ell$ ?

- Folklore: If  $e = \mathcal{O}(\log N)$ , then brute-force search runs in polynomial time.
- [GHM05]: If  $e \geq N^{1/4}$ , then as hard as factoring.

**Our result:**



## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$ed_p^{\text{MSB}} \approx e(d_p^{\text{MSB}} \| d_p^{\text{LSB}}) = 1 + k(p-1) \approx kp,$$

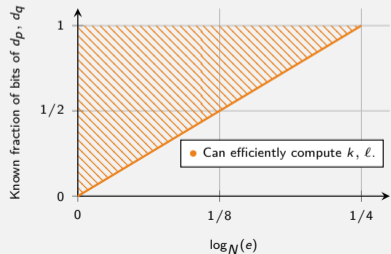
$$ed_q^{\text{MSB}} \approx e(d_q^{\text{MSB}} \| d_q^{\text{LSB}}) = 1 + \ell(q-1) \approx \ell q.$$

### Question

How difficult is computing  $k, \ell$ ?

- Folklore: If  $e = \mathcal{O}(\log N)$ , then brute-force search runs in polynomial time.
- [GHM05]: If  $e \geq N^{1/4}$ , then as hard as factoring.

**Our result:**



## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$ed_p^{\text{MSB}} \approx e(d_p^{\text{MSB}} \| d_p^{\text{LSB}}) = 1 + k(p-1) \approx kp,$$

$$ed_q^{\text{MSB}} \approx e(d_q^{\text{MSB}} \| d_q^{\text{LSB}}) = 1 + \ell(q-1) \approx \ell q.$$

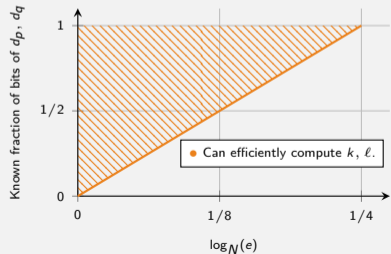
$$\implies e^2 d_p^{\text{MSB}} d_q^{\text{MSB}} \approx k\ell N.$$

### Question

How difficult is computing  $k, \ell$ ?

- Folklore: If  $e = \mathcal{O}(\log N)$ , then brute-force search runs in polynomial time.
- [GHM05]: If  $e \geq N^{1/4}$ , then as hard as factoring.

**Our result:**



## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$ed_p^{\text{MSB}} \approx e(d_p^{\text{MSB}} \| d_p^{\text{LSB}}) = 1 + k(p-1) \approx kp,$$

$$ed_q^{\text{MSB}} \approx e(d_q^{\text{MSB}} \| d_q^{\text{LSB}}) = 1 + \ell(q-1) \approx \ell q.$$

$$\implies e^2 d_p^{\text{MSB}} d_q^{\text{MSB}} \approx k\ell N.$$

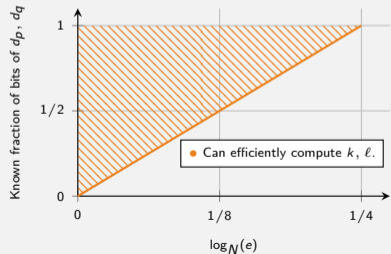
$$\implies \frac{e^2 d_p^{\text{MSB}} d_q^{\text{MSB}}}{N} \approx k\ell.$$

### Question

How difficult is computing  $k, \ell$ ?

- Folklore: If  $e = \mathcal{O}(\log N)$ , then brute-force search runs in polynomial time.
- [GHM05]: If  $e \geq N^{1/4}$ , then as hard as factoring.

**Our result:**



## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$ed_p^{\text{MSB}} \approx e(d_p^{\text{MSB}} \| d_p^{\text{LSB}}) = 1 + k(p-1) \approx kp,$$

$$ed_q^{\text{MSB}} \approx e(d_q^{\text{MSB}} \| d_q^{\text{LSB}}) = 1 + \ell(q-1) \approx \ell q.$$

$$\implies e^2 d_p^{\text{MSB}} d_q^{\text{MSB}} \approx klN.$$

$$\implies \frac{e^2 d_p^{\text{MSB}} d_q^{\text{MSB}}}{N} \approx kl.$$

### Lemma

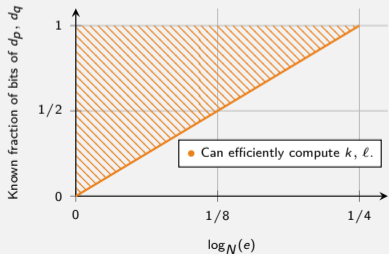
If  $d_p^{\text{MSB}}, d_q^{\text{MSB}} > e^2$ , then  $\left\lfloor \frac{e^2 d_p^{\text{MSB}} d_q^{\text{MSB}}}{N} \right\rfloor = kl.$

### Question

How difficult is computing  $k, \ell$ ?

- Folklore: If  $e = \mathcal{O}(\log N)$ , then brute-force search runs in polynomial time.
- [GHM05]: If  $e \geq N^{1/4}$ , then as hard as factoring.

**Our result:**





## Step 1: Compute Partial Solution in Few Variables

There exist  $k, \ell \in \mathbb{N}$ , such that

$$ed_p^{\text{MSB}} \approx e(d_p^{\text{MSB}} \| d_p^{\text{LSB}}) = 1 + k(p-1) \approx kp,$$

$$ed_q^{\text{MSB}} \approx e(d_q^{\text{MSB}} \| d_q^{\text{LSB}}) = 1 + \ell(q-1) \approx \ell q.$$

$$\implies e^2 d_p^{\text{MSB}} d_q^{\text{MSB}} \approx klN.$$

$$\implies \frac{e^2 d_p^{\text{MSB}} d_q^{\text{MSB}}}{N} \approx kl.$$

### Lemma

If  $d_p^{\text{MSB}}, d_q^{\text{MSB}} > e^2$ , then  $\left\lfloor \frac{e^2 d_p^{\text{MSB}} d_q^{\text{MSB}}}{N} \right\rfloor = kl$ .

### Lemma

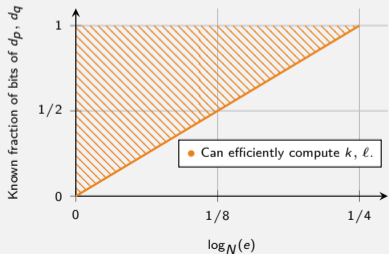
We can split  $kl$  into  $k$  and  $\ell$  in time  $\mathcal{O}(\log^2 N)$ .

### Question

How difficult is computing  $k, \ell$ ?

- Folklore: If  $e = \mathcal{O}(\log N)$ , then brute-force search runs in polynomial time.
- [GHM05]: If  $e \geq N^{1/4}$ , then as hard as factoring.

**Our result:**



## Step 2: Apply Coppersmith's Method

### Problem (Approximate GCD Problem)

Given:

- $N_0 = q_0 s$
- $N_1 \approx q_1 s$

Find:

- $s$

## Step 2: Apply Coppersmith's Method

### Problem (Approximate GCD Problem)

Given:

- $N_0 = q_0 s$
- $N_1 \approx q_1 s$

Find:

- $s$

### Theorem (Howgrave-Graham CaLC'01)

If  $s \geq N_0^\beta$ ,  $\beta \in [0, 1]$  and  $|N_1 - q_1 s| < N_0^{\beta^2}$ , then we can compute  $s$  in polynomial time.

- Algorithm is based on Coppersmith's method.

## Step 2: Apply Coppersmith's Method

### Problem (Approximate GCD Problem)

Given:

- $N_0 = q_0 s$
- $N_1 \approx q_1 s$

Find:

- $s$

### Theorem (Howgrave-Graham CaLC'01)

If  $s \geq N_0^\beta$ ,  $\beta \in [0, 1]$  and  $|N_1 - q_1 s| < N_0^{\beta^2}$ , then we can compute  $s$  in polynomial time.

- Algorithm is based on Coppersmith's method.

### Our attack scenario

Given:

- $N = qp$
- $ed_p^{\text{MSB}} \approx kp$

Find:

- $p$

## Step 2: Apply Coppersmith's Method

### Problem (Approximate GCD Problem)

Given:

- $N_0 = q_0 s$
- $N_1 \approx q_1 s$

Find:

- $s$

### Theorem (Howgrave-Graham CaLC'01)

If  $s \geq N_0^\beta$ ,  $\beta \in [0, 1]$  and  $|N_1 - q_1 s| < N_0^{\beta^2}$ , then we can compute  $s$  in polynomial time.

- Algorithm is based on Coppersmith's method.

### Our attack scenario

Given:

- $N = qp$
- $ed_p^{\text{MSB}} \approx kp$
- $k$

Find:

- $p$

## Step 2: Apply Coppersmith's Method

### Problem (Approximate GCD **Multiple** Problem)

Given:

- $N_0 = q_0 s$
- $N_1 \approx q_1 s$
- $q_1$

Find:

- $s$

### Theorem (Howgrave-Graham CaLC'01)

If  $s \geq N_0^\beta$ ,  $\beta \in [0, 1]$  and  $|N_1 - q_1 s| < N_0^{\beta^2}$ , then we can compute  $s$  in polynomial time.

- Algorithm is based on Coppersmith's method.

### Our attack scenario

Given:

- $N = qp$
- $ed_p^{\text{MSB}} \approx kp$
- $k$

Find:

- $p$

## Step 2: Apply Coppersmith's Method

### Problem (Approximate GCD **Multiple** Problem)

Given:

- $N_0 = q_0 s$
- $N_1 \approx q_1 s$
- $q_1$

Find:

- $s$

### Theorem (Howgrave-Graham CaLC'01)

If  $s \geq N_0^\beta$ ,  $\beta \in [0, 1]$  and  $|N_1 - q_1 s| < N_0^{\beta^2}$ , then we can compute  $s$  in polynomial time.

### Theorem

If  $s \geq N_0^\beta$ ,  $\beta \in [0, 1]$  and  $|N_1 - q_1 s| < q_1 N_0^{\beta^2}$ , then we can compute  $s$  in polynomial time.

### Our attack scenario

Given:

- $N = qp$
- $ed_p^{\text{MSB}} \approx kp$
- $k$

Find:

- $p$

## Step 2: Apply Coppersmith's Method

### Problem (Approximate GCD **Multiple** Problem)

Given:

- $N_0 = q_0 s$
- $N_1 \approx q_1 s$
- $q_1$

Find:

- $s$

### Theorem (Howgrave-Graham CaLC'01)

If  $s \geq N_0^\beta$ ,  $\beta \in [0, 1]$  and  $|N_1 - q_1 s| < N_0^{\beta^2}$ , then we can compute  $s$  in polynomial time.

### Theorem

If  $s \geq N_0^\beta$ ,  $\beta \in [0, 1]$  and  $|N_1 - q_1 s| < q_1 N_0^{\beta^2}$ , then we can compute  $s$  in polynomial time.

### Our attack scenario

Given:

- $N = qp$
- $ed_p^{\text{MSB}} \approx kp$
- $k$

Find:

- $p$

### Corollary

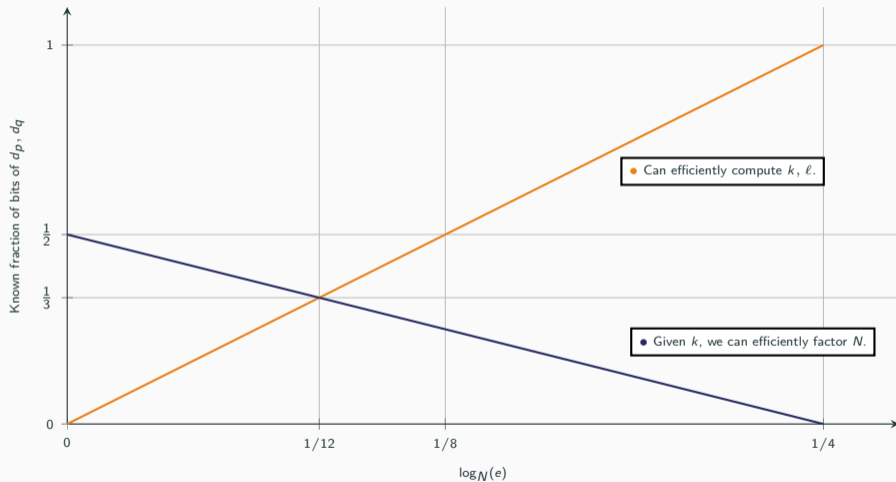
Given  $k$  and  $d_p^{\text{MSB}}$  with

$$d_p^{\text{MSB}} > \frac{N^{1/4}}{e},$$

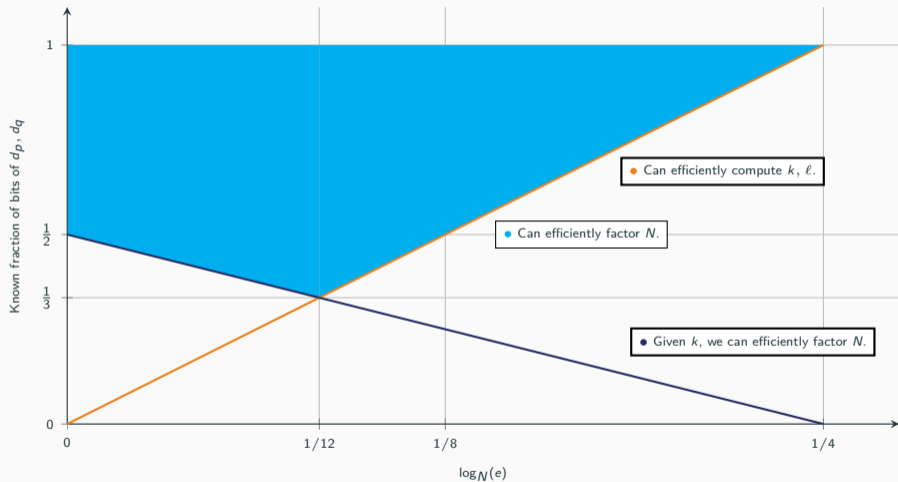
we can factor  $N$  in polynomial time.



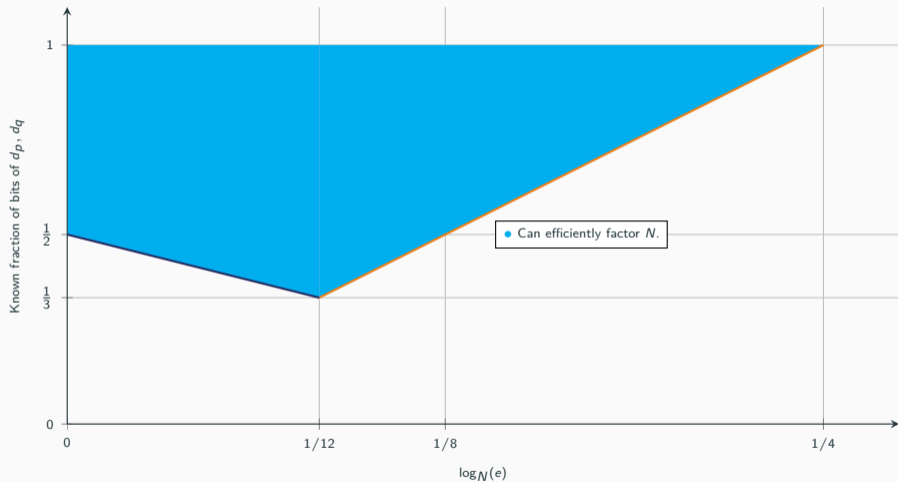
# Putting Everything Together



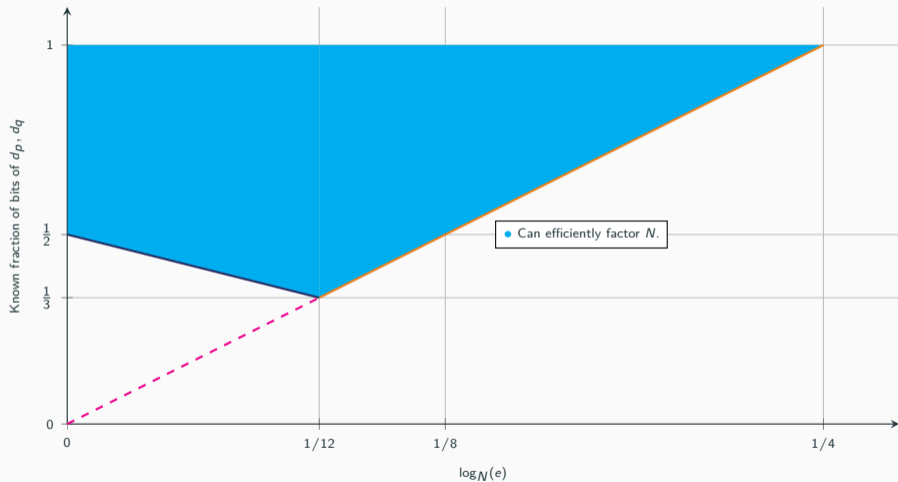
# Putting Everything Together



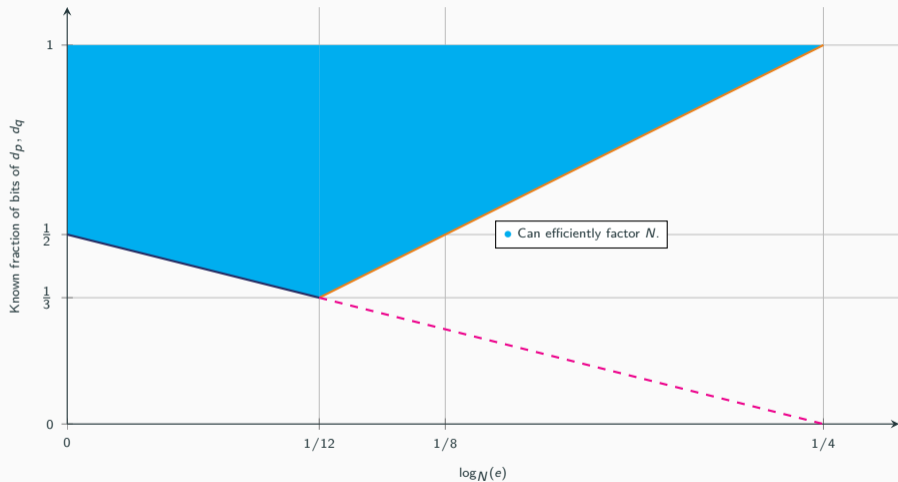
# Putting Everything Together



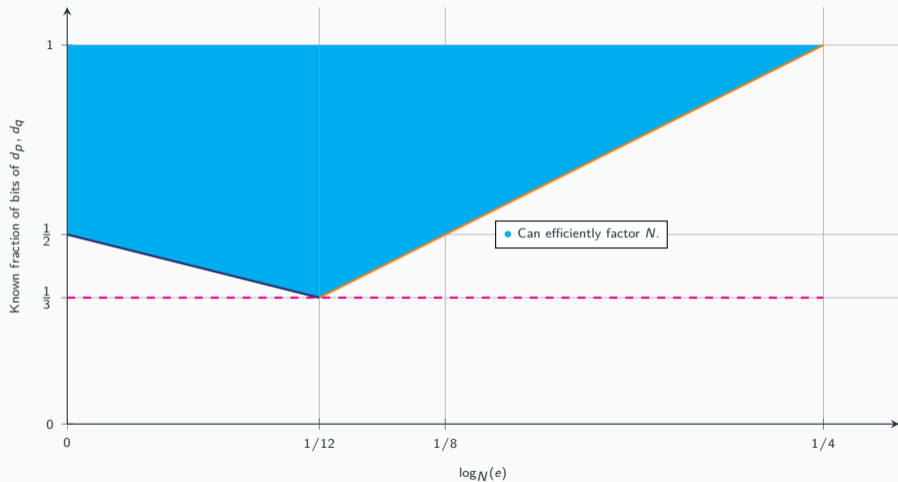
# Putting Everything Together



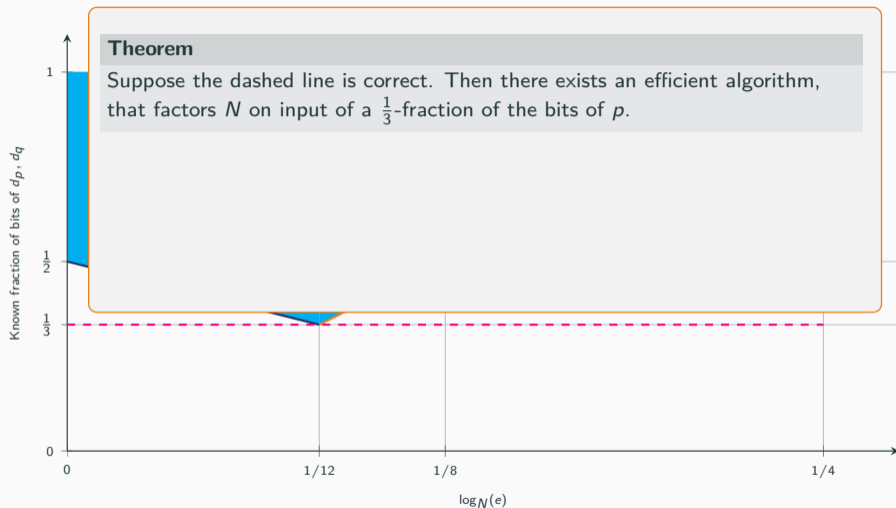
# Putting Everything Together



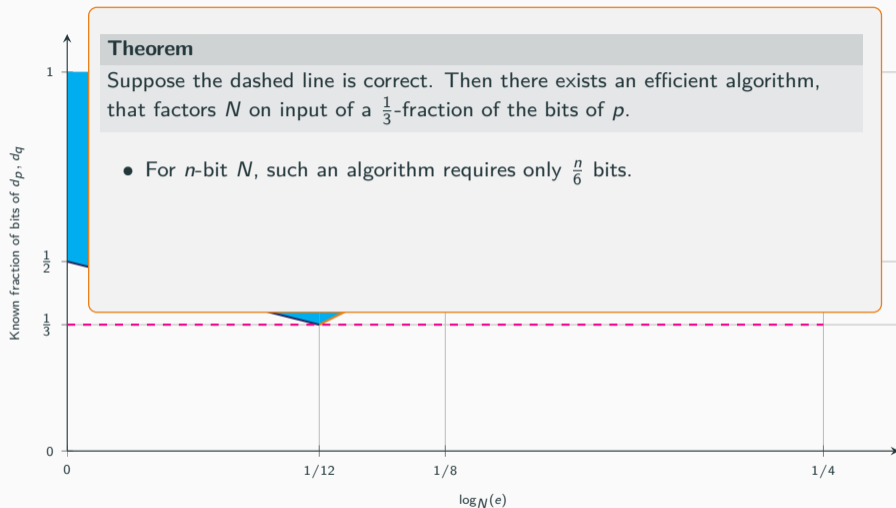
# Putting Everything Together



# Putting Everything Together

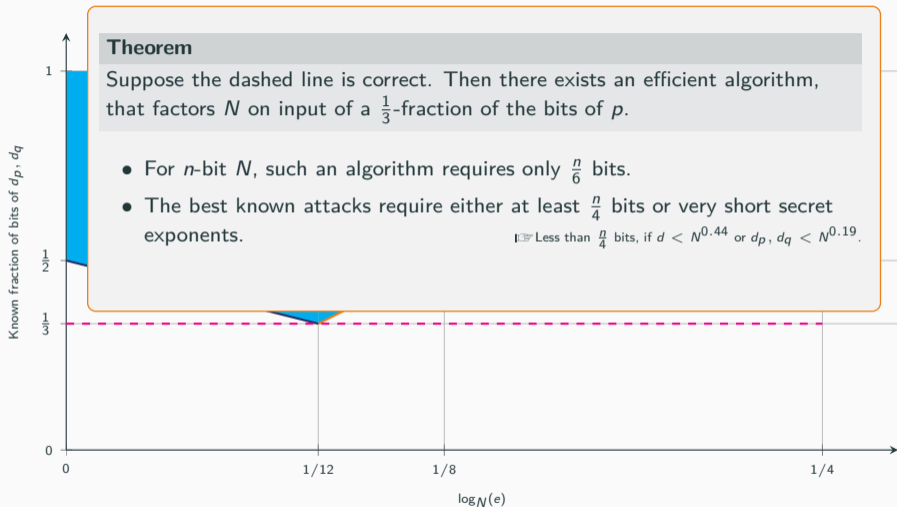


# Putting Everything Together





# Putting Everything Together



### Conclusion:

- Previously known Partial Key Exposure attacks work the better, the smaller  $e$ ,  $d$ ,  $d_p$ ,  $d_q$ .
- First Partial Key Exposure attack on RSA, with a different behavior.
- Works best for  $e \approx N^{1/12}$ .

### Conclusion:

- Previously known Partial Key Exposure attacks work the better, the smaller  $e$ ,  $d$ ,  $d_p$ ,  $d_q$ .
- First Partial Key Exposure attack on RSA, with a different behavior.
- Works best for  $e \approx N^{1/12}$ .
- Take-away: Do not apply Coppersmith's method directly to your system of polynomial equations. Check first, if you can eliminate some variables by different means.

### Conclusion:

- Previously known Partial Key Exposure attacks work the better, the smaller  $e$ ,  $d$ ,  $d_p$ ,  $d_q$ .
- First Partial Key Exposure attack on RSA, with a different behavior.
- Works best for  $e \approx N^{1/12}$ .
- Take-away: Do not apply Coppersmith's method directly to your system of polynomial equations. Check first, if you can eliminate some variables by different means.

### Open Problems:

- Which size of  $e$  should we use in practice?
- Is  $e \approx N^{1/12}$  the least secure?

## Conclusion:

- Previously known Partial Key Exposure attacks work the better, the smaller  $e$ ,  $d$ ,  $d_p$ ,  $d_q$ .
- First Partial Key Exposure attack on RSA, with a different behavior.
- Works best for  $e \approx N^{1/12}$ .
- Take-away: Do not apply Coppersmith's method directly to your system of polynomial equations. Check first, if you can eliminate some variables by different means.

## Open Problems:

- Which size of  $e$  should we use in practice?
- Is  $e \approx N^{1/12}$  the least secure?
- Does our algorithm for the AGCD-Multiple-Problem have implications for the AGCD-Problem?

# Comparison Between Partial Key Exposure Attacks

Exposed variable	Constraint	Required bits
$p$	-	$\frac{n}{4}$
$d$	$e = \mathcal{O}(\log N)$	$\frac{n}{4}$
$d_p$	$e = \mathcal{O}(\log N)$	$\frac{n}{4}$
$d$	$d < N^{0.44}$	$< \frac{n}{4}$
$d$	$d < N^{0.36}$	$< \frac{n}{8}$
$d$	$d < N^{0.29}$	0
$d_p, d_q$	$d_p, d_q < N^{0.29}$	$< 2 \times \frac{n}{4}$
$d_p, d_q$	$d_p, d_q < N^{0.19}$	$< 2 \times \frac{n}{8}$
$d_p, d_q$	$d_p, d_q < N^{0.12}$	0
$d_p, d_q$	$e \leq N^{1/8}$	$\leq 2 \times \frac{n}{4}$
$d_p, d_q$	$e \approx N^{1/12}$	$2 \times \frac{n}{6}$