Attacks against RSA and its Implementations

Arne Ansper, Ahto Buldas

RSA

- Invented by Rivest, Shamir and Adleman in 1977
- Secret key: prime numbers p, q; and private exponent d
- Public key: modulus $N=pq$ and public exponent e
- $ed \bmod (p-1)(q-1) = 1$, i.e. $ed = 1 + k\,(p-1)(q-1)$
- Encryption of a message M: $C = M^e \bmod N$
- Decryption (signing) of C: $C^d \bmod N = M^{ed} \bmod N = M$

Secure Encryption

- Ciphertext C must not reveal any information about the plaintext M *(semantic security)*
- The “textbook RSA” is not semantically secure
- Example, encrypting yes/no votes. Given an encrypted vote $C = v^e \bmod N$, an attacker can easily encrypt both votes and compare the results to C.
- Random padding has to be applied before encryption

Secure Signatures

- Existential unforgeability: Given message/signature pairs

  $(M_1,S_1), (M_2,S_2), \dots , (M_m, S_m)$

  it must be impossible to create one more signature (M,S)
- “Textbook RSA” is not existentially unforgeable, because of the homomorphic property:

  $M_1^d\, M_2^d \bmod N = (M_1 M_2)^d \bmod N$

- Paddings have to be used!

Classification: Targets

- Against RSA itself: factoring large integers, quantum computers and Shor’s algorithm
- Against improper use of RSA in protocols: common modulus, blinding
- Against improper choice of parameters: low private exponent, low public exponent, Hastad broadcast attack, Franklin-Reiter related message attack, Coppershmith’s short pad attack, etc.
- Against improper implementations: partial key exposure attacks, improper random numbers, timing-attacks, power-consumption attacks, random faults, Bleichenbacher’s attack on PKCS#1 padding

Classification: Inputs

- Public-key only attacks
- Attacks that require physical access to implementation

Classification: Impact

- Decrypted ciphertext
- Forged signature
- Factorization of the modulus
- Method of factoring all moduli (e.g. Shor’s algorithm)

Partial Key Exposure

- Given an n-bit RSA modulus N, and $n/4$ least significant bits of the secret modulus d, it is easy to compute d
- Given an n-bit RSA modulus $N=pq$, and $n/4$ least/most significant bits of p, the modulus N can be factored (Coppersmith 1996)

Timing Attacks

- Let $d_n d_{n-1} \ldots d_1 d_0$ be the bit-representation of d. The computation of M^d mod N is performed as follows:

  $z := M$, $C := 1$
  For every $i = 0 \ldots N-1$ do:
  	if $d_i = 1$, then $C := C\,z$ mod N
  	$z := z^2$ mod N

- The attacker asks the smartcard to compute a large number of exponents, measures the times and reconstructs d using statistical analysis.

Random Faults

- Smartcard applications of RSA frequently use CRT (Chinese Reminder Theorem) to speed up M^d mod N where $N=pq$

  $d' := d \bmod p-1$ $\quad d'' := d \bmod q-1$
  $C' := M^{d'} \bmod p$ $\quad C'' := M^{d''} \bmod q$
  $C := c'q\,C' + c''p\,C'' \bmod N$ where c' and c'' are constants such that $c'q + c''p = 1$

- Error occurs when computing C'' and $\underline{C}$ is the erroneous version of C. Then

  $\underline{C}^e = M \bmod p$ but $\underline{C}^e \neq M \bmod q$

- Hence, attacker can compute gcd (N, $\underline{C}^e - M$) = p and to factorize N

CYBERNETICA

Bleichenbacher’s Attack

- The PKCS 1 padding looks like as follows:

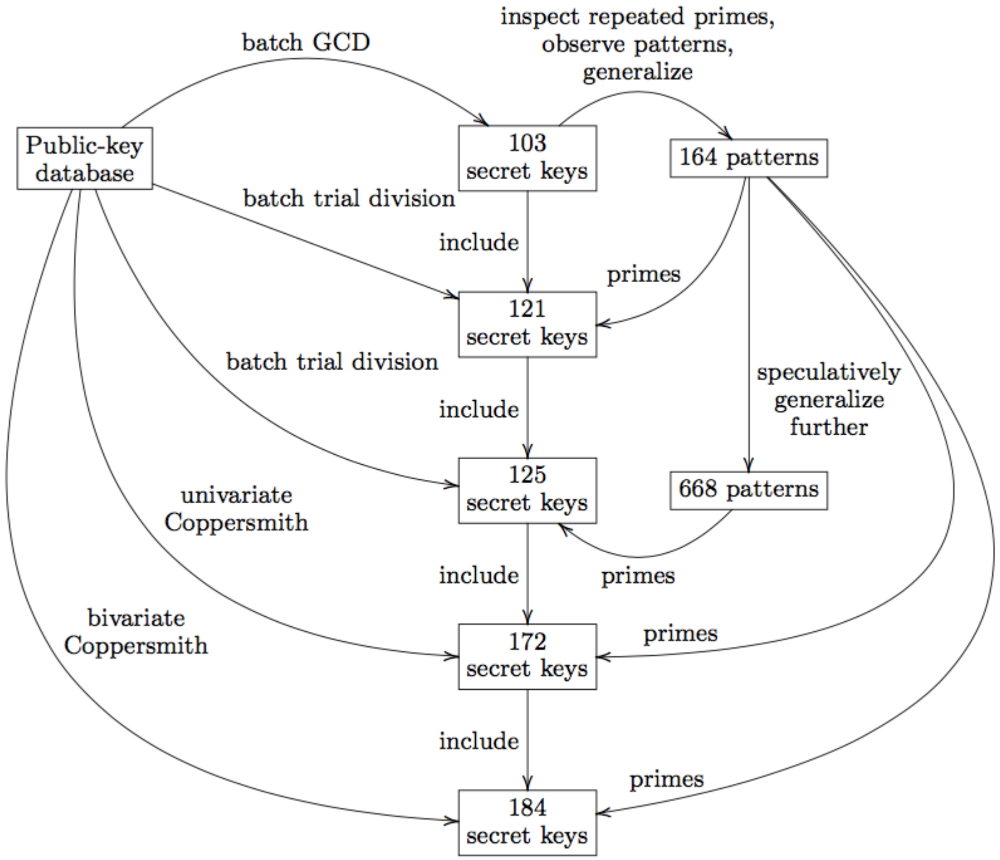
  02 | Random | 00 | Message

- Say a server receives encrypted messages and returns an “invalid ciphertext” error message if the decrypted message has an incorrect padding
- So, sending a “random” ciphertext C to the server, an attacker will know if the corresponding plaintext has 02 in the beginning
- Bleichenbacher showed in 1998 that if an attacker who has access to such a server, can decrypt any ciphertext

Weak Random Numbers

- Cryptographically secure random numbers are crucial for generating proper RSA keys
- In 2012: Lenstra et al discovered that many public-key certificates contain the same public keys and many share a common prime
- In 2013: Taiwan’s secure digital IDs use weak random
- Out of about 2 million 1024-bit RSA keys,184 keys were generated so poorly they could be broken in a matter of hours. Some pairs of keys shared the same prime number

Attack against Taiwan digital IDs



Daniel J. Bernstein, Yun-An Chang, Chen-Mou Cheng, Li-Ping Chou, Nadia Heninger, Tanja Lange, and Nicko van Someren. 2013. Factoring RSA Keys from Certified Smart Cards: Coppersmith in the Wild. In *Advances in Cryptology - ASIACRYPT 2013*. Springer-Verlag, 341–360.

Weak Prime Numbers

- Even if the random numbers used by a smartcard are ok, the choice of prime numbers may be poor
- That is the case with the current Estonian ID-card incident
- To search for prime numbers, smart-cards use several (so called) fast-prime methods, that fasten the search, but at the same time, reduce the number of candidate primes
- Each such method characterizes the card and can be identified by running tests on public moduli
- We can check if a key is produced by the particular Infineon chip used by the Estonian ID-card

CYBERNETICA

The Weakness

- Nemec, Sys, Svenda, Klinec, and Matyas discovered that the Infineon chip produces prime numbers of the form:

  $$p = 65537^a \bmod M + kM,$$

  where M is constant and the same for all chips. For 2048-bit modulus N, it is the product of the first 126 primes.
- Hence, all public moduli N satisfy $(65537^c\text{-}N) \bmod M$. This is *the test* of weak moduli, the authors published.
- Naive search: try all $\mathrm{ord}_M(65537)$ possible a-s and try to find k
- *Naive search does not work*: the number of a-s to examine would be 2^{254}

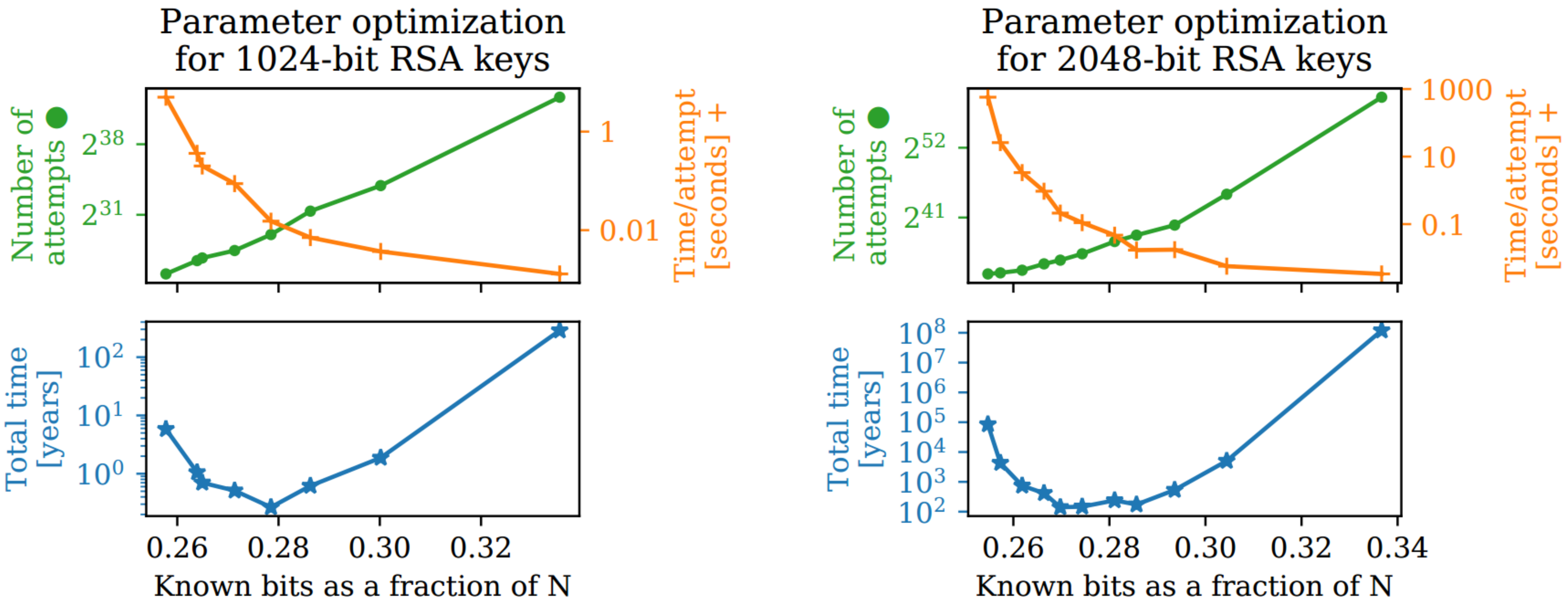
CYBERNETICA

The Science

- **Main idea**: Use a divisor *M'* of *M*, such that $\mathrm{ord}_{M'}(65537)$ is feasible, but still the number of bits in *M'* is larger than 2048/4 (necessary for the Coppersmith attack)
- Then, the prime numbers are still expressible in the form:

$$p = 65537^{a'} \bmod M' + k'M'$$

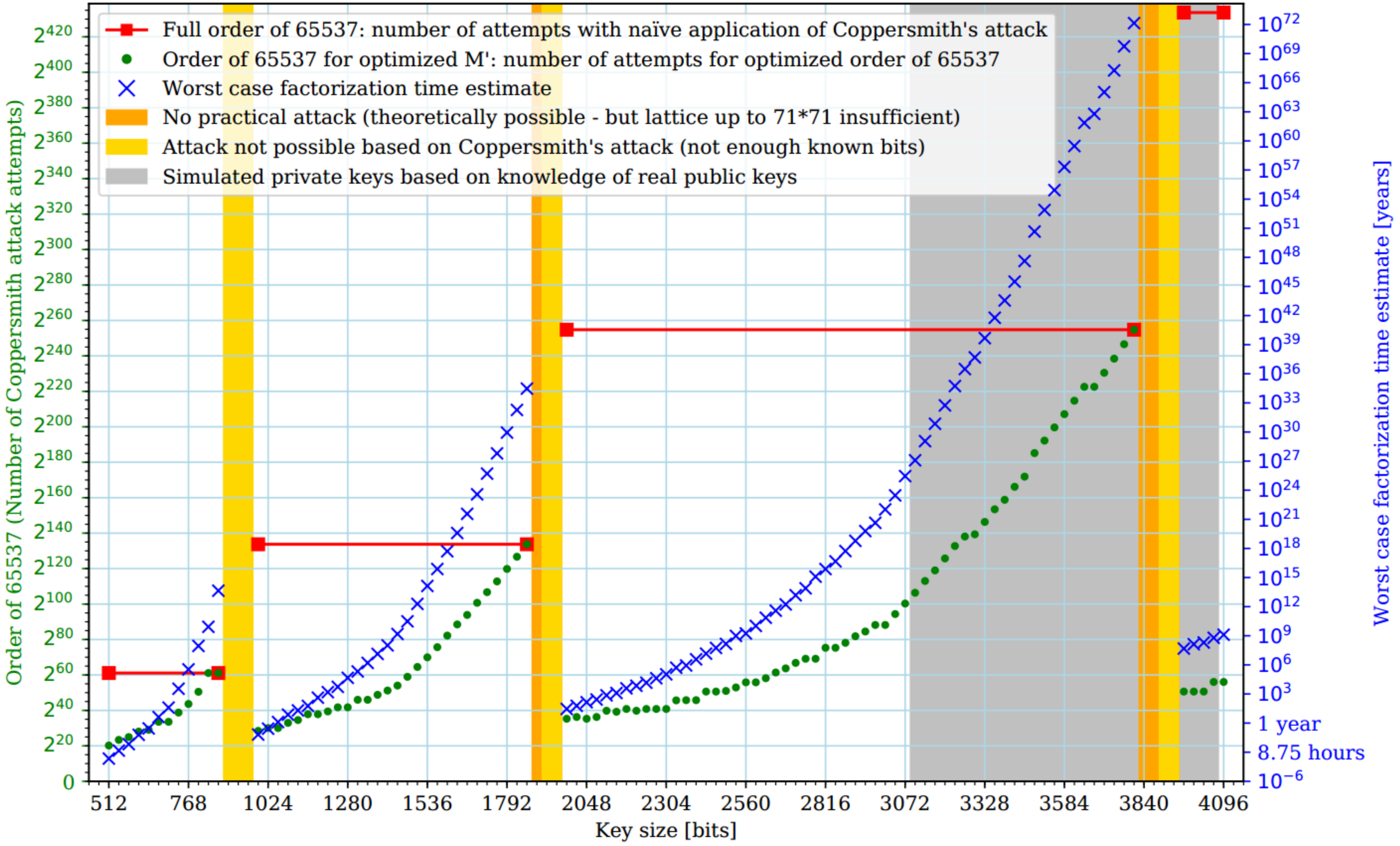
- Authors found optimal *M'* in terms of the overall attack time by brute force search combined with greedy heuristics



Matus Nemec and Marek Sys and Petr Svenda and Dusan Klinec and Vashek Matyas. 2017. The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli. In *24th ACM Conference on Computer and Communications Security (CCS'2017)* ACM, 1631-1648

The Impact

- By using optimal *M'*, the number of possible *a*-s is 2^{34} for 2048-bit RSA modulus
- *k* is found in 200 ms (on one core of 3GHz Intel Xeon E5-2650 v3) by using the Coppersmith's algorithm (1996)
- The total cost per key is about 140 CPU years



Matus Nemec and Marek Sys and Petr Svenda and Dusan Klinec and Vashek Matyas. 2017. The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli. In *24th ACM Conference on Computer and Communications Security (CCS'2017)* ACM, 1631-1648

CYBERNETICA

Certified is not Secure

- The Infineon chip is Common Criteria certified
- How could such a flaw slip through certification?
- Certifications do not certify that the product is secure against known and unknown threats
- They just certify that certain functions were implemented according to specification
- At the time of certification, the fast prime generation methods were not known to have any weaknesses

CYBERNETICA

Conclusions

- In spite of having been attacked through 40 years, RSA itself has no known weaknesses
- Only quantum computers can break RSA efficiently
- Vulnerabilities in soft- and hardware are inevitable
- We must have mechanisms in place to mitigate those vulnerabilities
- IT-Systems design/management must take potential unknown vulnerabilities into account

CYBERNETICA