Attacks

https://iuk.one/1033-1321

Clemens H. **Cap**
ORCID: 0000-0003-3958-6136

Department of Computer Science
University of **Rostock**
Rostock,Germany
clemens.cap@uni-rostock.de

Version 2

Overview

1. Some Attacks on RSA
2. Bellcore Attack
3. Fault Attacks

Attacks Revisited

Goal:

- Closer look at attacks using the example of RSA
- Discovering the Bellcore attack
- Learning about fault attacks

Non-Goal:

- Complete overview on attacks
- This has been done elsewhere and especially in lecture “Datensicherheit”

1. Some Attacks on RSA

Standard algorithms may have a wide range of known attacks.

We demonstrate this using the example of RSA.

1. Some Attacks on RSA

2. Bellcore Attack

3. Fault Attacks

Shared Modulus Attack

Attack:

- If two parties share the value of the modulus $n = p{\cdot}$q
- and both parties have a private key and a public key,
- then one party can derive the private key of the other party from its public key.
- Knowing n, e and d allows to factor n.

Protection:

- Do not intentionally share the modulus
- Use proper randomization

Low Public Exponent (Coopersmith) Attack (1)

Situation: Often a low public exponent is used.

Motivation:

- Speeds up encryption process
- Also: Use Fermat primes
  Form $2^{2^x} + 1$, is prime for many x, but not for $x = 5$
- Common choices: 3, 17, 65537
- Effect: They are close to a power of 2, so recursive exponentiation is fast

Attack:

- Several attacks are known for this scenario.
- When messages are short.
- When several messages are encrypted and differ only by a small value.

Low Public Exponent (Coopersmith) Attack (2)

Protection:

- Do not use small public exponent.
- Problem: Some libraries still select small exponents.
- Problem: Some texts still recommend small exponents.
- Use proper padding
  Effect: Prevents attacks capitalizing on small message sizes
  effect: Prevents differential attacks capitalizing on small message differences

Low Private Exponent (Wiener) Attack

Attack:

- Similar as Coopersmith attack, just for fast decryption performance.
- Moreover: If $d < n^{1/4}/3$ there is a possibility for an attacker to determine d.

Protection:

- Do not use small private exponent

Quantum Computer Attacks

Grover Algorithm:

- If a quantum computer can be built
- with a sufficient number of qubits
- larger than available today (2023),
- then factoring is easy.

Bleichenbacher Attack (1)

Attack:

- Adaptive chosen ciphertext attack on SSL handshaking phase
- in combination with a particular padding scheme.
- Client is supposed to handshake with encrypted message which has a particular padding
- Attacker does not know key and starts with a particular message
- Server informs client on the particular nature of the error (padding / crypto error)
- Attacker adjusts the ciphertext and retries
- Also known as "millions attack" as it needs large number of attempts by client.

Bleichenbacher Attack (2)

Protection:

- Server should not inform client on particular nature of error
- Recommended: Drop detailed error messages when switching from dev to production
- Use a more randomized padding which is less predictable for the involved parties (eg. OAEP)
- If software notices suspicious behavior it should blacklist the source for some time

  It must not continue to provide an oracle service for an attacker.

Shared Primefactor Attack

Attack:

- If moduli n_1 and n_2 of two parties share a prime factor, both parties are compromised.
- $gcd(n_1, n_2)$ yields a common prime factor, division the second prime factors.

Protection:

- Use good sources of randomness for key generation.

Sidechannel Attacks

Power Attack:

- **Attack:** Power consumption varies with CPU bus weights.
  Power consumption allows reconstruction of keys.
- **Protection:** Shield system from power consumption measurement (battery)

Acoustic Attack:

- **Attack:** CPU capacitor noise provides info on power consumption.
  Then continue with power attack.
- **Protection:** Shield system from acoustic emanations.

Timing Attack:

- **Attack:** Duration of computation varies with key and message structure.
  Timing information allows reconstruction of keys.
- **Protection:** Provide response always after the same fixed time.

Fault Attacks

Wide range of attacks possible.

Own chapter, to look at those.

Specific example: Bellcore attack.

2. Bellcore Attack

Astonishing form of a fault attack.

The one example of all the attacks, which we shall study in more detail.

1. Some Attacks on RSA
2. Bellcore Attack
3. Fault Attacks

Chinese Remainder Theorem (CRT)

Let $n_1, \dots, n_k$ be pairwise coprime integers with $\forall i \colon n_i > 1$.
Let $a_1, \dots, a_k$ be integers such that $\forall i : 0 \leq a_i < n_i$.
Consider the k equations $x \equiv a_1 \bmod n_1 \quad \dots \quad x \equiv a_k \bmod n_k$.

Then the following holds:

1. There exists **exactly one** solution x_0 satisfying all these k equations with
$$0 \leq x_0 \leq n_1 \cdot \ldots \cdot n_k - 1$$
2. The **set of all** solutions x satisfying all these k equations may be written in the form
$$\{x_0 + z \cdot (n_1 \cdot \ldots \cdot n_k) \mid z \in \mathbb{Z}\}$$
3. These k equations are equivalent to the **single equation**
$$x \equiv x_0 \bmod n_1 \cdot \ldots \cdot n_k$$

We do the proof for $k = 2$.

Proof: Existence of Solution

Let n_1 and n_2 be coprime.

The $k = 2$ equations read $x \equiv a_1 \mod n_1$ and $x \equiv a_2 \mod n_2$.

Then $gcd(n_1, n_2) = 1$.

Use extended Euclidean algorithm and Bezout identity to get m_1, m_2 such that

$$m_1 \cdot n_1 + m_2 \cdot n_2 = gcd(n_1, n_2) = 1$$

This also means

$$m_1 \cdot n_1 \equiv 1 \mod n_2 \text{ and } m_2 \cdot n_2 \equiv 1 \mod n_1$$

Thus m_1 is inverse of n_1 modulo n_2 and similar for m_2.

Construct a solution of the $k = 2$ equations by setting

$$x_0 = a_1 m_2 n_2 + a_2 m_1 n_1$$

Verify solution property by direct computation using above formulae.

Proof: Uniqueness of Representation

Let r and s be solutions of $x \equiv a_1 \bmod n_1$ and $x \equiv a_2 \bmod n_2$.

Then $(r - s) \equiv 0 \bmod n_1$ and $(r - s) \equiv 0 \bmod n_2$.

Thus there exist l_1 and l_2 such that $(r - s) = l_1 n_1$ and $(r - s) = l_2 n_2$.

This means $(r - s) = l_1 n_1 = l_2 n_2$

Since n_1 and n_2 are coprime, there even exists l such that $(r - s) = l n_1 n_2$.

To see this, collect the (different) prime numbers
in the prime number decompositions of n_1 and n_2 of $(r - s)$.

The difference between two solutions r and s thus is a multiple of $n_1 n_2$.

This demonstrates the uniqueness.

Proof: Equivalent Equation

Let $x \equiv a_1 \bmod n_1$ and $x \equiv a_2 \bmod n_2$ with n_1 and n_2 coprime.

In the proof of existence we found a solution:

$$x_0 = a_1 m_2 n_2 + a_2 m_1 n_1$$

We found that differences between solutions are multiples of $n_1 n_2$.

An equivalent single modulo equation thus is:

$$x \equiv \underbrace{a_1 m_2 n_2 + a_2 m_1 n_1}_{\text{One solution}} \quad \underbrace{\bmod n_1 n_2}_{\text{differences between solutions}}$$

Application of Chinese Remainder Theorem

Task: We want to compute $x^d \bmod pq$.

Precompute:

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

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

m_1 as multiplicative inverse of d_p

m_2 as multiplicative inverse of d_q

We know by the theorem of Fermat: $x^{\phi}(n) \equiv 1 \bmod n$. Thus:

$$x^d \equiv x^{d_p} \bmod p$$

$$x^d \equiv x^{d_q} \bmod q$$

Bellcore Attack (1)

Let $n = pq$ be a modulus with corresponding e and d.

We have a message x.

We let the opponent sign the message x.

We assume he uses the CRT for signing.

This means we precompute s_1 and s_2 and solve for the unknown x^d the system

$$x^d \equiv s_1 \mod p$$

$$x^d \equiv x^d \equiv \quad \mod q$$

using the CRT.

This means we combine s_1 and s_2 into a signature value s.

Bellcore Attack (2)

We now obtain the same signature for a second time.

The opponent repeats the precomputation steps.

The precomputation values this time are s_1' and s_2'.

This time we provoke faults in the signing device of the opponent.

We assume the fault kicks in at
exactly one of the two CRT computations steps of s_1', s_2'.

We assume this happens with s_1'. This means we assume

$$x^d \not\equiv s_1' \bmod p$$

$$x^d \equiv s_2' \bmod q$$

Now we combine s_1' and s_2' into signature s'.

Bellcore Attack (3)

Since the fault occurred in prime factor p we obtain

$$s \not\equiv s' \mod p$$

$$s \equiv s' \mod q$$

Thus $s - s' \not\equiv 0 \mod p$ and $s - s' \equiv \mod q$.

Can we learn something from the value of $s - s' \mod p$?

Probably not, as the difference might depend heavily on the particular fault.

Can we learn something from $s - s' \equiv 0 \mod q$?

Probably not, as this is expected all the time in CRT signing.

Bellcore Attack (4)

But wait!

Normally we have

$$s - s' \equiv 0 \bmod q \qquad s - s' \equiv 0 \bmod p \qquad s - s' \equiv 0 \bmod pq$$

Now we have

$$s - s' \equiv 0 \bmod q \qquad s - s' \not\equiv 0 \bmod p \qquad s - s' \not\equiv 0 \bmod pq$$

Can we use the fact that $s - s' \equiv 0 \bmod q$ in a context where $s - s' \not\equiv \ \bmod pq$?

Bellcore Attack (5)

We have $s - s' \equiv 0 \bmod q$ and thus $s - s' = \lambda q$.

Let us calculate $gcd(s - s', pq) = gcd(\lambda qmpq)$.

λq does not have p as a factor since $s - s' \not\equiv 0 \bmod p$.

Since p and q are primess: $gcd(s - s', pq) = q$,

So we can factor n.

Launching and Counteracting the Bellcore Attack

Requirements:

- Need 2 signatures
- Need that CRT is used
- Need possibility to inject faults along a particular fault model

Countermeasures:

- Check signature before it leaves the device.
- Use different implementation.
- Use several rounds of blinding and unblinding to prevent precise injection of error in one step.
- Monitor device and halt processing on first malfunction.

3. Fault Attacks

What are fault attacks in general?

What other examples are there?

How can we protect against them?

1. Some Attacks on RSA
2. Bellcore Attack
3. Fault Attacks

Fault Model (1)

What is the systematic perspective of the attacker?

Granularity

- How many bits are affected by the fault?
- Single bit error
- Few bits error
- Big change

Modification

- Stuck-at-zero or stuck-at-one
- Flip
- Random

Fault Model (2)

Control on Fault

- Precise control of location (bit) and timing (when) of error
- Can activate and deactivate error at will any time
- Smaller amount of control on error
- No control at all, error is completely random

Duration of Fault

- Full control (can activate and deactivate)
- Transient (takes place for some time, but cannot control)
- Permanent (turn it on and it stays for the rest of the devices life time)

Categories of Fault Injection

Non-invasive: Not necessary to open / damage chip package

Soft: Slow modification of operational limits introduces some random faults

- Eg: Slow change in operating temperature introduces some random faults
- Eg: Slow change in clock speed may provoke particular faults as soon as design timing parameter limits are reached
- Eg: Slow change in voltage of power supply

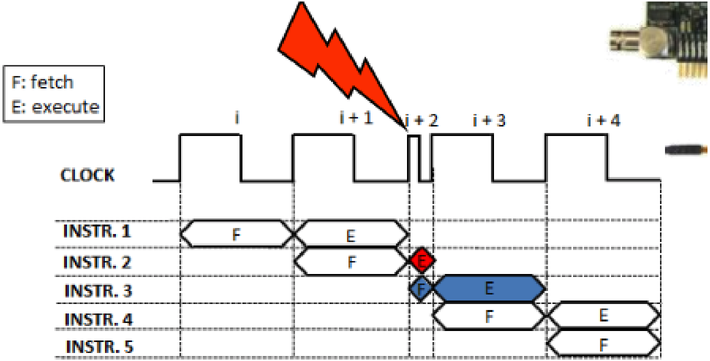
Targeted: Modification of operational limits with targeted effects

Semi-invasive: Chip decapsulation

Invasive: Need electrical contact to chip

Example: Targeted Clock Manipulation

Attack: A carefully crafted temporal overclocking can prevent certain portions of an algorithm execute properly.

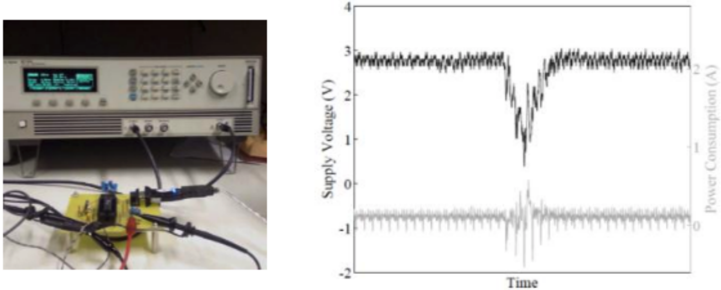


[BGV11]

Fig. 1: Targeted clock Manipulation

Example: Targeted Voltage Manipulation

Attack: A carefully crafted manipulation of the supply voltage can cause certain parts of a system to fail at certain moments.



[KQ07]

[SH08]

Fig. 2: Targeted Voltage Manipulation

Example: Soft Thermal Manipulation

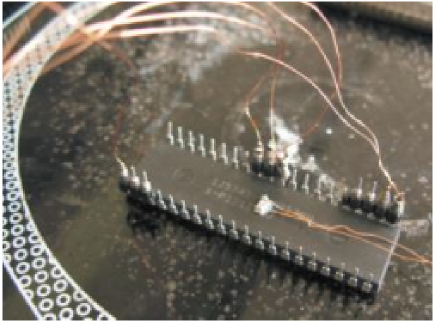


Fig. 3: Chip slowly heated from below

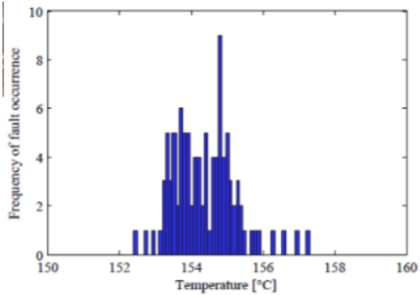


Fig. 4: Fault occurrence depending on temperature

Example: Optical Fault Injection

Attack:

- Semiconductors are sensitive to light
- Optical pulses can cause a transistor to switch
- Fault injection by shining a flash into a chip
- Fault injection by directing a laser beam on to a chip

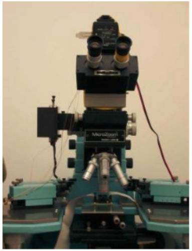
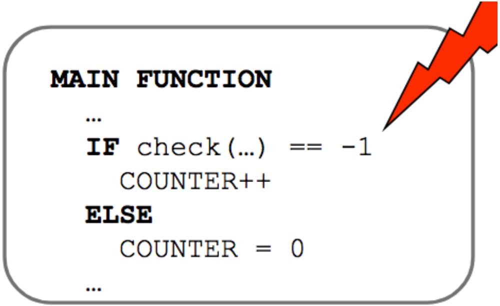


Fig. 5: Workbench for optical fault injection

Example: Attack on a retry Counter

Attack:

- Manipulate a retry counter value change
- Block arithmetic unit
- Prevent change of the retry counter



```
MAIN FUNCTION
  ...
  IF check(...) == -1
    COUNTER++
  ELSE
    COUNTER = 0
  ...
```

Fig. 6: Fault attack on a retry counter

Strategy for Checks

Recommended strategy for checks:

- Disable the access system
- Do a check
- If check is ok, enable the system again

Not safe:

- Do a check
- If check is not ok, disable the system

More Complex Fault Attacks

Protection:

- Check the different levels of the crypto stack
- At every layer a fault attack is possible

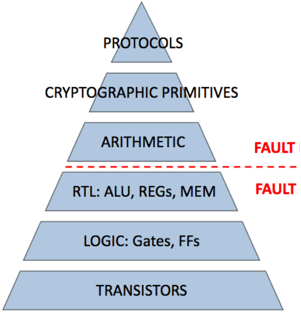


Fig. 7: Layers on which a fault attack seems possible

Differential Attacks

Idea: Compare output of crypto algorithm

- normal output
- output under small changes to input
- output under small injected faults

Example: Bellcore is a differential attack.

Countermeasures: Hardening Hardware (1)

Hide sensitive parts of the chip

- Raise the price for the attacker
- Security by obscurity is state of the art here
- Bus scrambling to make signal tracking more difficult
- Superfluous logic elements to make reengineering more difficult
- Additional shiedling to prevent injection of impulse

Operational constraint sensors

- Let chip detect violation of operational constraints
- Shut down chip if clock, power etc. is outside limits

Countermeasures: Hardening Hardware (2)

Tampering sensors

- Let chip detect if it is watched
- Light sensors for detecting case openings

Destruct upon opening architectures:

- Removal of metal layers leads to removal of logic attacker wants to analyze

Further Countermeasures (1)

Add plausibility checks into computation

Parallel execution and result comparison

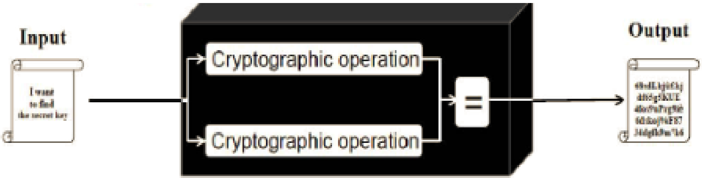


Fig. 8:

Further Countermeasures (2)

Inverse execution and test upon completion

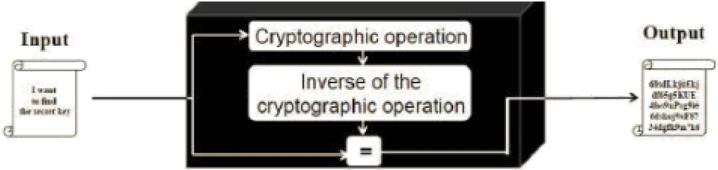


Fig. 9:

Branchless implementation

- Branches allow easy interruption of control flow

Appendix

Contents of Appendix

List of Figures

Terms of Use (1)

Die hier angebotenen Inhalte unterliegen deutschem Urheberrecht. Inhalte Dritter werden unter Nennung der Rechtsgrundlage ihrer Nutzung und der geltenden Lizenzbestimmungen hier angeführt. Auf das Literaturverzeichnis wird verwiesen. Das Zitatrecht in dem für wissenschaftliche Werke üblichen Ausmaß wird beansprucht. Wenn Sie eine Urheberrechtsverletzung erkennen, so bitten wir um Hinweis an den auf der Titelseite genannten Autor und werden entsprechende Inhalte sofort entfernen oder fehlende Rechtsnennungen nachholen. Bei Produkt- und Firmennamen können Markenrechte Dritter bestehen. Verweise und Verlinkungen wurden zum Zeitpunkt des Setzens der Verweise überprüft; sie dienen der Information des Lesers. Der Autor macht sich die Inhalte, auch in der Form, wie sie zum Zeitpunkt des Setzens des Verweises vorlagen, nicht zu eigen und kann diese nicht laufend auf Veränderungen überprüfen.

Alle sonstigen, hier nicht angeführten Inhalte unterliegen dem Copyright des Autors, Prof. Dr. Clemens Cap, ©2020. Wenn Sie diese Inhalte nützlich finden, können Sie darauf verlinken oder sie zitieren. Jede weitere Verbreitung, Speicherung, Vervielfältigung oder sonstige Verwertung außerhalb der Grenzen des Urheberrechts bedarf der schriftlichen Zustimmung des Rechteinhabers. Dieses dient der Sicherung der Aktualität der Inhalte und soll dem Autor auch die Einhaltung urheberrechtlicher Einschränkungen wie beispielsweise Par 60a UrhG ermöglichen.

Die Bereitstellung der Inhalte erfolgt hier zur persönlichen Information des Lesers. Eine Haftung für mittelbare oder unmittelbare Schäden wird im maximal rechtlich zulässigen Ausmaß ausgeschlossen, mit Ausnahme von Vorsatz und grober Fahrlässigkeit. Eine Garantie für den Fortbestand dieses Informationsangebots wird nicht gegeben.

Die Anfertigung einer persönlichen Sicherungskopie für die private, nicht gewerbliche und nicht öffentliche Nutzung ist zulässig, sofern sie nicht von einer offensichtlich rechtswidrig hergestellten oder zugänglich gemachten Vorlage stammt.

Use of Logos and Trademark Symbols: The logos and trademark symbols used here are the property of their respective owners. The YouTube logo is used according to brand request 2-9753000030769 granted on November 30, 2020. The GitHub logo is property of GitHub Inc. and is used in accordance to the GitHub logo usage conditions `https://github.com/logos` to link to a GitHub account. The Tweedback logo is property of Tweedback GmbH and here is used in accordance to a cooperation contract.

Terms of Use (2)

Disclaimer: Die sich immer wieder ändernde Rechtslage für digitale Urheberrechte erzeugt ein nicht unerhebliches Risiko bei der Einbindung von Materialien, deren Status nicht oder nur mit unverhältnismäßig hohem Aufwand abzuklären ist. Ebenso kann den Rechteinhabern nicht auf sinnvolle oder einfache Weise ein Honorar zukommen, obwohl deren Leistungen genutzt werden.

Daher binde ich gelegentlich Inhalte nur als Link und nicht durch Framing ein. Lt EuGH Urteil 13.02.2014, C-466/12 (Pressemitteilung, Blog-Beitrag, Urteilstext). ist das unbedenklich, da die benutzten Links ohne Umgehung technischer Sperren auf im Internet frei verfügbare Inhalte verweisen.

Wenn Sie diese Rechtslage stört, dann setzen Sie sich für eine Modernisierung des völlig veralteten Vergütungs- und Anreizsystems für urheberrechtliche Leistungen ein. Bis dahin klicken Sie bitte auf die angegebenen Links und denken Sie darüber nach, warum wir keine für das digitale Zeitalter sinnvoll angepaßte Vergütungs- und Anreizsysteme digital erbrachter Leistungen haben.

Zu Risiken und Nebenwirkungen fragen Sie Ihren Rechtsanwalt oder Gesetzgeber.

Weitere Hinweise finden Sie im Netz hier und hier oder hier.

Citing This Document

If you use contents from this document or want to cite it,
please do so in the following manner:

Clemens H. Cap: Attacks. Electronic document. https://iuk.one/1033-1321 4. 7. 2023.

Bibtex Information: https://iuk.one/1033-1321.bib

```
@misc{doc:1033-1321,
   author      = {Clemens H. Cap},
   title       = {Attacks},
   year        = {2023},
   month       = {7},
   howpublished = {Electronic document},
   url         = {https://iuk.one/1033-1321}
}
```

Typographic Information:
Typeset on ?today?
This is pdfTeX, Version 3.14159265-2.6-1.40.21 (TeX Live 2020) kpathsea version 6.3.2
This is pgf in version 3.1.5b
This is preamble-slides.tex myFormat©C.H.Cap

List of Slides

Legend:

⧉ continuation slide

○ slide without title header

🖼 image slide