Attacks on RSA (3)

Timing Attacks

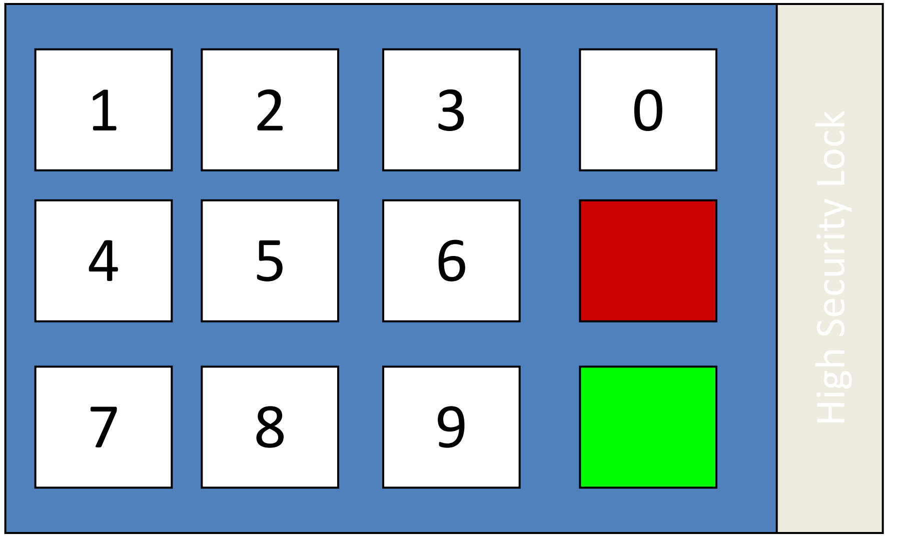
General Overview

- Focus on timing attacks on RSA
- What is a timing attack?
- What might cause timing leaks?
- Simple exploitation of leaks
- Differential timing attack

Beware: in order to follow the lectures you NEED to be familiar with various cryptographic algorithms and implementation techniques!

A Simple Timing Attack applied to a Combination Lock

- Assume the lock takes an n-digit code and checks efficiently whether it is correct, i.e.:
  - Digits are checked one after another
  - As soon as a wrong digit has been entered the red light goes on
  - Only if all n digits were entered correclty the green light goes on



- Apparently the time until a light goes on depends on the the correctness of each digit
  - The more digits are correct the longer it takes
  - Code can be easily (less than brute force) discovered

Red light: wrong combination entered

Green light: correct combination entered

Simple timing analysis, cont.

- The previous example was very simple but
  - If you program something that checks the correctness of a certain combination, would you not also check each item in the combination?
  - Would you not also try write efficient code?
  - A large number of access systems did the checking of the codeword in this manner (who knows how many still do...)
- A `simple’ countermeasure
  - Ensure that the response time is fixed
- **A first conclusion**
  - **Defending against such attacks requires to write less efficient code**

Timing attacks against RSA

- We now investigate how such attacks apply to RSA (what might cause timing leaks?)
- We also have a first look at how we can extend the princle of simple side channel analysis to differential side channel analysis:
  - First we focus on timing variations that can be exploited from one single measurement
    - This timing information relates to the `type‘ of intermediate operation
  - Then we focus on timing variations that require many measurements to be exploited
    - This timing information relates to the actual value of an intermediate variable

Sources for timing side channels

- Different operations have different timings:
  - Gate level: AND gates doesn‘t need to wait for other inputs if one is zero
  - Square of a number might be quicker to compute than the multiplication of two different numbers
  - Loops take longer the more iterations there are in those loops
  - A conditional branching might induce different timings
  - If Montgomery multiplication/squaring is used then the final subtraction might induce different timings

Simple timing attack on RSA

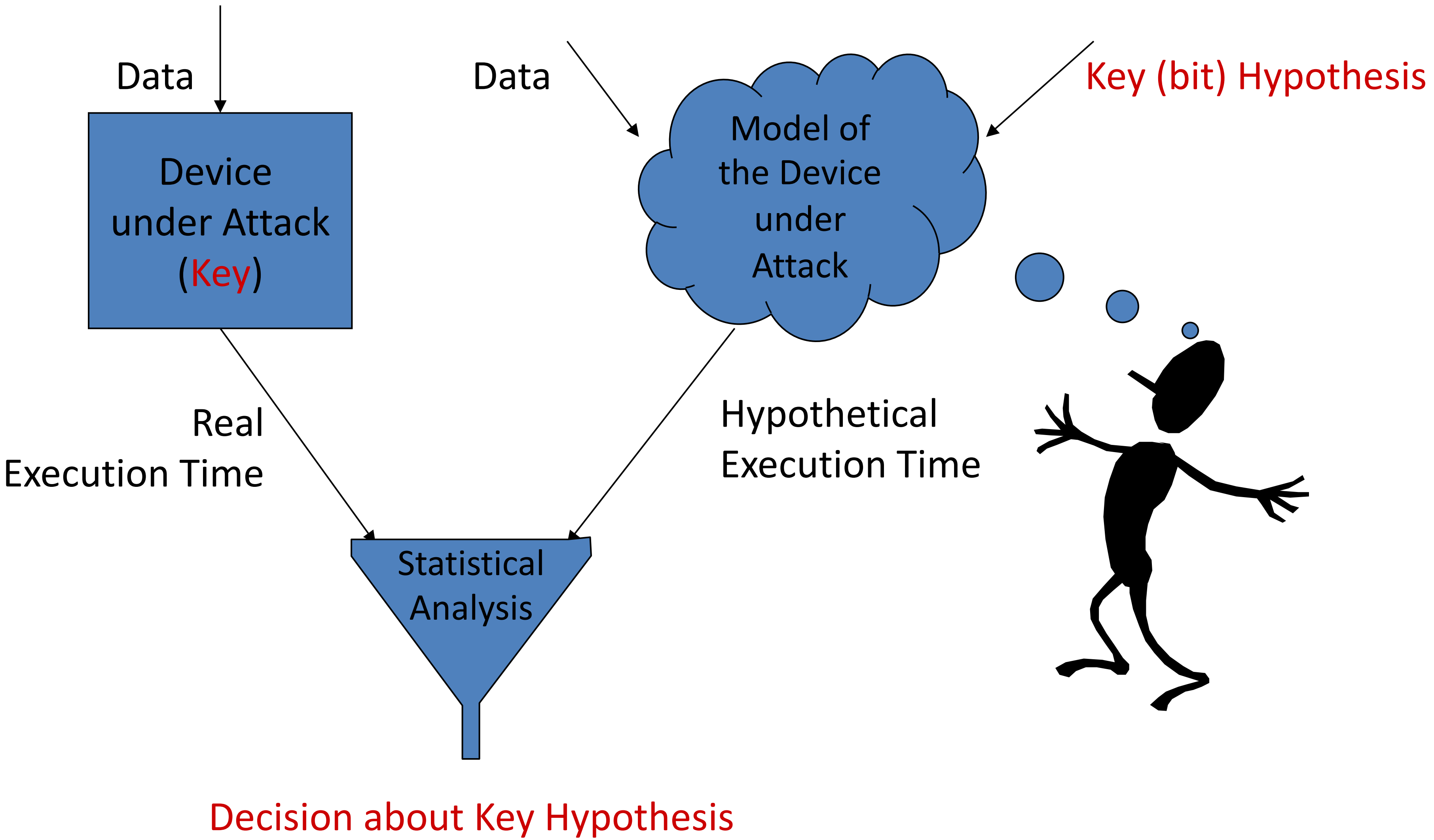
- The overall duration reveals the number of loop iterations, i.e. the lenght of the secret key

- Then in step i a multiplication is only performed iff d_i =1
  - Timing depends on the bits of the key
  - Hence the overall timing also reveals the the Hamming weight of the key
- Can we do any better than this??

$d=\{d_w,d_{w-1},d_{w-2},\ldots,d_1,d_0\}_2$

```
m = 1;
For i = w-1 to 0
  m = m • m mod n
  if (d_i) == 1
    then m = m • c mod N
  (endif)
(endfor)
```

Principle of a differential timing attack



Principle of a differential timing attack, in words ...

- Choose a set of ciphertexts
- Model:
  - guess one (few) bit(s) of the key
    - We call these the key hypotheses
  - calculate one (few) iteration(s) of the square and multiply algorithm
  - predict the execution times per ciphertext
    - Hypothetical execution time

- Device: decrypt the same set of ciphertexts
- Analysis: compare the hypothetical timing of the model with the actual execution time
  - If similar then key hypothesis was correct

Differential timing attack, practical example for RSA, using the paper by Dhem et al.

Device under attack

$d=\{d_w,d_{w-1},d_{w-2},\ldots,d_1,d_0\}_2$

m = 1

For i = w-1 to 0

  m = m • m mod n

  if (d_i) == 1

    then m = m • c mod N

  (endif)

(endfor)

Model (have already i-1 bits of key)

$H_{i,0}=\{d_w,d_{w-1},\ldots,d_{w-i-1},0\}_2$
$H_{i,1}=\{d_w,d_{w-1},\ldots,d_{w-i-1},1\}_2$

m = 1

For j = w-1 to i

  m = m • m mod n

  if ($H_{j,b}$) == 1

    then m = m • c mod N

  (endif)

(endfor)

A Practical Implementation of the Timing Attack, Dhem et al., 1998, available from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.3070

Differential timing attack, exploiting Montgomery reduction

- Montgomery arithmetic:
  - Montgomery arithmetic has a constant execution time (for a certain number of bits) but
  - For some values an extra subtraction is required at the end. This is what we can exploit in a differential timing attack.

```
m = 1;
For j = w-1 to 0
  m = m • m mod n /*(MontMul) */
  if (d_j) == 1
    then m = m • c mod N /*(MontMul)*/
  (endif)
(endfor)
```

Algorithm ($\mathbb{Z}_N$-MontMul)

Input: Two base-b, unsigned integers, $0 \leq x, y < N$

Output: A base-b, unsigned integer $r = x \cdot y \cdot \rho^{-1} \pmod{N}$

```
r ← 0
for i = 0 upto l_N − 1 step +1 do
  | u ← (r_0 + y_i · x_0) · ω (mod b)
  | r ← (r + y_i · x + u · N)/b
end
if r ≥ N then
  | r ← r − N
end
return r
```

Differential timing attack, MR exploited at multiplication

Device under attack

$d=\{d_w,d_{w-1},d_{w-2},\ldots,d_1,d_0\}_2$

m = 1

For i = w-1 to 0

  m = m • m mod n

  if (d_i) = 1

    then m = m • C mod N

  (endif)

(endfor)

Model

$H_i=\{d_w,d_{w-1},\ldots,d_{w-i-1},0\}_2$ (i=0)
$H_i=\{d_w,d_{w-1},\ldots,d_{w-i-1},1\}_2$ (i=1)

m = 1

For j = w-1 to i

  m = m • m mod n

  if (H_j) == 1

    then m = m • C mod N

  (endif)

(endfor)

Differential timing attack, MR exploited at multiplication

Device under attack

if (d_i) = 1

then m = m • c mod N

Assume d_i=1: then the multiplication is carried out. A MR might be required (or not).

Model

if (H_i) == 1

then m = m • c mod N

If $H_i = d_i$ the model behaves like the real device (up until bit i) and it will like the real device carry out a MR (or not).

If $H_i <> d_i$ the model does NOT behave like the real device.

Informally: By analysing lots of timings from the device and comparing these for every bit hypothesis to the averages of the two model options we can eventually determine which model option (i.e. key hypothesis) is correct. This reveals the key bit by bit.

DTA, RSA: Distinguisher

- More formally described we are seeking to evaluate the difference or ranking of a 2-element statistical quantity called distinguishing vector

$$\hat{D}_N(k)_{k\in\{0,1\}} = \left\{\hat{D}_N\left(T\left(k^*,x\right),M\left(k,x\right)\right)\right\}_{k\in\{0,1\}}$$

  – With N ... Number of texts, x being a set of inputs, k_i* the correct key hypothesis, k all key hypothesis in the key hypothesis space K={0,1}

- An attack successful identifies k_i* iff

$$\hat{D}_N(k_i^*) > \hat{D}_N(k)\forall k \in \mathrm{K}$$

An example with ‘numbers’ …

$$\hat{D}_N(H_i)_{H_i\in\{0,1\}} = \left\{\hat{D}_N\left(T\left(k^*,x\right), M(H_i,x)\right)\right\}_{H_i\in\{0,1\}}$$

$$= \left|\operatorname{avg}\left(T \middle| M(H_i,x) = \text{red}\right) - \operatorname{avg}\left(\text{T} \middle| M(H_i,x) = \text{no red}\right)\right|$$

Example were H_i=1 is correct

T	Red?	H_i=0	H_i=1	D_N(1)
12	Yes	-	Yes	12
14	Yes	-	Yes	13
8	No	-	No	5
12	Yes	-	Yes	4.67
15	Yes	-	Yes	5.25
10	No	-	No	4.25

- Assume that previous i-1 key bits are already known
  - So next bit is attacked, we assume it to be 1
  - We take 6 measurements (listed under T)
  - The column called Red shows if or not in the real device a reduction took place
- The column H_i=1 shows the model predictions, D_N(1) the distinguisher outcome

An example with ‘numbers’ …

$$\hat{D}_N(H_i)_{H_i\in\{0,1\}} = \left\{\hat{D}_N\left(T\left(k^*,x\right), M(H_i,x)\right)\right\}_{H_i\in\{0,1\}}$$

$$= \left|\ \mathrm{avg}\left(T \middle| M(H_i,x) = \mathrm{red}\right) - \mathrm{avg}\left(\mathrm{T} \middle| M(H_i,x) = \mathrm{no\ red}\right)\right|$$

- Now an example where we hypothesise d_i to be one BUT it actually is zero
- The predictions do NOT correspond to the behaviour of the device

Example were H_i=1 is incorrect

T	Red?	H_i=0	H_i=1	$D_N(1)$
12	No	-	Yes	12
14	Yes	-	No	2
8	No	-	No	4
12	Yes	-	Yes	3
15	Yes	-	Yes	3.66
10	No	-	No	3.66

An example with ‘numbers’ …

$$\hat{D}_N(H_i)_{H_i\in\{0,1\}} = \left\{\hat{D}_N\left(T\left(k^*,x\right),M\left(H_i,x\right)\right)\right\}_{H_i\in\{0,1\}}$$

$$= |\operatorname{avg}\left(T \middle| M(H_i,x)=\text{red}\right) - \operatorname{avg}\left(\mathrm{T} \middle| M(H_i,x)=\text{no red}\right)|$$

Example were H_i=1 is correct

T	Red?	H_i=0	H_i=1	$D_N(1)$
12	Yes	-	Yes	12
14	Yes	-	Yes	13
8	No	-	No	5
12	Yes	-	Yes	4.67
15	Yes	-	Yes	5.25
10	No	-	No	4.25

Example were H_i=1 is incorrect

T	Red?	H_i=0	H_i=1	$D_N(1)$
12	No	-	Yes	12
14	Yes	-	No	2
8	No	-	No	4
12	Yes	-	Yes	3
15	Yes	-	Yes	3.66
10	No	-	No	3.66

For a correct key guess the distinguisher shows a higher value than for an incorrect key guess, because the model predictions correspond to the behaviour of the real device.

Another attack strategy (again using the Dhem et al. Paper)

- The previous attack was not great as we had to find a ‘baseline’ for distinguisher values (i.e. what is ‘high’ vs. ‘low’)
- A better way to proceed is to attack the square ‘in the next round’

$H_i=\{d_w,d_{w-1},\ldots,d_{w-i-1},0\}_2$ (i=0)
$H_i=\{d_w,d_{w-1},\ldots,d_{w-i-1},1\}_2$ (i=1)

```
m = 1
For j = w-1 to i
   m = m • m mod n
   if (Hj) == 1 /* key guess */
      then m = m • C mod N
   (endif)
(endfor)
m = m • m mod n
```

i

i+1

Another example with 'numbers' …

$$\hat{D}_N(H_i)_{H_i\in\{0,1\}} = \left\{\hat{D}_N\left(T\left(k^*,x\right),M\left(H_i,x\right)\right)\right\}_{H_i\in\{0,1\}}$$

$$=|\operatorname{avg}\left(T\,\middle|\,M(H_i,x)=\text{red}\right)-\operatorname{avg}\left(\mathrm{T}\,\middle|\,M(H_i,x)=\text{no red}\right)|$$

Example were H_i=1 is correct

T	Red?	H_i=0	H_i=1	$D_N(0)$	$D_N(1)$
12	Yes	Yes	Yes	12	12
14	Yes	No	Yes	2	13
8	No	No	No	1	5
12	Yes	No	Yes	0.66	4.67
15	Yes	Yes	Yes	2.16	5.25
10	No	Yes	No	1.08	**4.25**

Flow for H_i=1

```
m = 1
For j = w-1 to i-1
  m = m • m mod n
  if (H_j) == 1
    then m = m • C mod N
  (endif)
(endfor)
m = m • m mod N
m = m^2 • C mod N
m = m^2 • C • m mod N
```

Flow for H_i=0

```
m = 1
For j = w-1 to i-1
  m = m • m mod n
  if (H_j) == 1
    then m = m • C mod N
  (endif)
(endfor)
m = m • m mod N
m = (m^2)^2 mod N
```

Another example with ‘numbers’ …

$$\hat{D}_N(H_i)_{H_i \in \{0,1\}} = \left\{ \hat{D}_N \left(T\left(k^*, x\right), M\left(H_i, x\right) \right) \right\}_{H_i \in \{0,1\}}$$

$$= \left| \operatorname{avg}\left(T \middle| M(H_i, x) = \text{red} \right) - \operatorname{avg}\left(\mathrm{T} \middle| M(H_i, x) = \text{no red} \right) \right|$$

Example were H_i=1 is correct

T	Red?	H_i=0	H_i=1	$D_N(0)$	$D_N(1)$
12	Yes	Yes	Yes	12	12
14	Yes	No	Yes	2	13
8	No	No	No	1	5
12	Yes	No	Yes	0.66	4.67
15	Yes	Yes	Yes	2.16	5.25
10	No	Yes	No	1.08	**4.25**

Example were H_i=0 is correct

T	Red?	H_i=0	H_i=1	$D_N(0)$	$D_N(1)$
12	No	No	Yes	12	12
14	Yes	Yes	No	2	1
8	No	No	No	4	1
12	Yes	Yes	Yes	3	1
15	Yes	Yes	Yes	3.66	2
10	No	No	No	**3.66**	2.33

Another example with a correlation distinguisher

$$\hat{D}_N(k)_{k\in \mathrm{K}} = \left\{ \hat{D}_N\left(T\left(k_i^*, x\right), M\left(k, x\right)\right)\right\}_{k\in \mathrm{K}} = |\,\mathrm{R(T,M)}\,|$$

$$R(T,M) = \frac{Cov(T,M)}{\sqrt{Var(T)Var(M)}} = \frac{E(T\cdot M) - E(T)E(M)}{\sqrt{Var(T)Var(M)}}$$

$$= \frac{E(T - E(T))E(M - E(M))}{\sqrt{Var(T)Var(M)}}$$

- Correlation is a natural way of measuring similarity between data
- But we need some way to model „red“ and „no red“
  - Could be with 0 and 1, or with proper timings

Example cont.

H ={0 … NR,1 … R }

T	Red?	H_i=0	H_i=1	$D_N(0)$	$D_N(1)$
12	Yes	1	1		
14	Yes	0	1		
8	No	0	0		
12	Yes	0	1		
15	Yes	1	1		
10	No	1	0	0.21	**0.85**

M={1 … no red,3 … red}

T	Red?	H_i=0	H_i=1	$D_N(0)$	$D_N(1)$
12	No	1	3		
14	Yes	3	1		
8	No	1	1		
12	Yes	3	3		
15	Yes	3	3		
10	No	1	1	**0.78**	0.49

Correlation distinguisher cont.

- Quality of model crucial for disinction between correct and incorrect key guess
- Ways to derive timing model
  - Stick to simplest: 0/1
    - This just means that there is some difference
  - Use educated guess
    - Based on your knowledge of the system
  - Use available data with known inputs
    - Try and derive the duration of a reduction on average
  - Profiling
    - For sets of key bits/plaintext derive precicely what timings would be

Summary

- We looked at different timing attacks on RSA
  - Simple vs. Differential timing attacks (aka single observation vs. Many observations)
- A differential timing attack compares real and simulated timings to derive information about the correct key hypothesis
  - This is done using distinguishers
    - Averages aka distinance of means test
    - Correlation analysis (using a suitable timing model)
- You can now get going with the timing attack assignment!