Timing Attacks against RSA
(Data Security and Privacy)

Lorenzo Palloni

University of Florence

lorenzo.palloni@stud.unifi.it

February 1, 2022

Introduction

- Main project goal $\rightarrow$ overview of two major timing attacks:
  - Kocher's timing attack (1996) [1];
  - Brumley and Boneh's timing attack (2005) [2].
- A timing attack is a type of side-channel attack.
- A side-channel attack exploits physical parameters, such as:
  - execution time;
  - electromagnetic emission;
  - supply current.

Square-and-multiply modular exponentiation

Algorithm 1 Square-and-multiply modular exponentiation algorithm.

function mod_exp(y, x, n) ▷ Computes y^x mod n
    $R \leftarrow 1$
    for $k \leftarrow 0, w-1$ **do**
        $R \leftarrow (R \cdot R)$ mod n
        if (the k-th bit of x) is 1 **then**
            $R \leftarrow (R \cdot y)$ mod n
    return R

mod_exp is a core operation in public-key cryptosystems, such as:

- RSA;
- Diffie-Hellman key exchange.

Kocher's timing attack - assumptions

Assume an RSA cryptosystem, with $D_k[x] = y^x \bmod n$.
Now, suppose that an attacker:

- wants to retrieve the private exponent x;
- already knows the first b exponent bits of x;
- can perform as many decryption operations as he wants;
- is able to measure $T := e + \sum_{i=0}^{k-1} t_i$, where:
  - $t_i \rightarrow i$-th iteration required time for $\text{mod\_exp}(y, x, n)$;
  - $y \rightarrow$ any ciphertext;
  - $e \rightarrow$ overhead time.

Kocher's timing attack - pseudocode

Algorithm 2 Kocher's timing attack

1: generate s ciphertexts $\{y_1, \ldots, y_s\}$;
2: guess the b-th exponent bit $d'_b := 0$;
3: measure $T'_j = e + \sum_{i=0}^{b-1} t'_i, \qquad \forall j \in \{1, \ldots, s\}$;
4: estimate $\mathrm{Var}(T - T')$;
5: repeat from step 3. and step 4. with $d'_b := 1$;
6: choose $d^*_b \in \{0, 1\}$ that minimizes $\mathrm{Var}(T - T')$;
7: set $d_b \leftarrow d^*_b$;
8: set $b \leftarrow b + 1$;
9: repeat from step 2. until $b > k - 1$.

In step 3., T' is measured by running mod_exp(y, x_b, n), where:

- $x_b := (d_0 d_1 \cdots d_{b-1} d'_b)_2$.

Brumley and Boneh's timing attack - assumptions

Assume an RSA cryptosystem implemented with OpenSSL (0.9.7).
Let $n = pq$ be the public modulus, with $q < p$.
Now, suppose that an attacker:

- wants to retrieve the private factor q;
- knows $i-1$ bits of q: $\{q_0, q_1, \dots, q_{i-1}\}$;
- starts to guess q with g:
  - $\rightarrow\ g_0 := q_0,\ g_1 := q_1,\ \dots,\ g_{i-1} := q_{i-1};$
  - $\rightarrow\ g_i := 0,\ g_{i+1} := 0,\ \dots,\ g_{k-1} := 0;$
- can perform as many decryption operations as he wants;
- knows that his guess $g \in [2^{511}, 2^{512}-1]$ [1].

[1] The public modulus n in OpenSSL 0.9.7 has a 1024-bit binary representation.

Brumley and Boneh's timing attack - pseudocode

Algorithm 3 Brumley and Boneh's timing attack against OpenSSL

1: set $g' := g$, then $g'_i := 1$;
2: compute $u_g = gR^{-1} \bmod n$, and $u_{g'} = g'R^{-1} \bmod n$;
3: measure $t_1 = \text{decryption\_time}(u_g)$, and $t_2 = \text{decryption\_time}(u_{g'})$;
4: compute $\Delta = |t_1 - t_2|$;
5: **return** 0 if Δ is "large". Otherwise (Δ is "small"), **return** 1.

Note that in step 1., if $q_i = 1$:

- then $g < g' < q$;
- otherwise, $g < q < g'$.

Brumley and Boneh's timing attack - why it works

Techniques implemented in OpenSSL (0.9.7) to improve mod_exp:

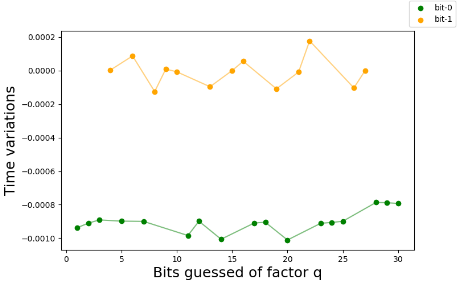
- Chinese Remainder $\rightarrow$ exposes $q \Rightarrow p = n/q \Rightarrow d = e^{-1} \bmod \phi(n)$ [2];
- Sliding Windows [5] $\rightarrow$ many multiplications by g;
- Montgomery multiplication [4] $\rightarrow$ more time required when $g < q$ [6];
- Karatsuba's algorithm $\rightarrow$ less time required when g approaches q from below.

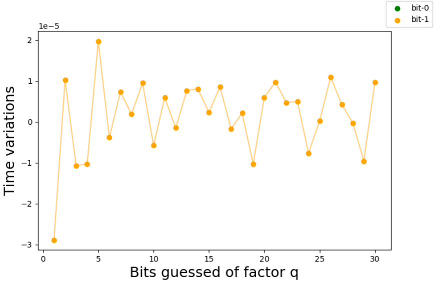
[2] $\phi(n) = \phi(p) \cdot \phi(q) = (p-1) \cdot (q-1)$.

Brumley and Boneh's timing attack - simulations

Time variations for each guessed bit (32-bits factor):

- without blinding (left): almost deterministic;
- with blinding (right): unpredictable.





Conclusion

In 1996, Kocher:

- showed that simple mod_exp exposes the exponent;
- prompted improvements on mod_exp implementations.

In 2005, Brumley and Boneh:

- proved that remote timing attacks are practical;
- made crypto libraries to implement blinding by default.

Thanks for your attention!

Do you have any questions?

References

- Kocher, P.C., 1996, August. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Annual International Cryptology Conference (pp. 104-113). Springer, Berlin, Heidelberg.
- Brumley, D. and Boneh, D., 2005. Remote timing attacks are practical. Computer Networks, 48(5), pp.701-716.
- Boreale, M., 2003. Note per il corso di Sicurezza delle Reti.
- Montgomery, P.L., 1985. Modular multiplication without trial division. Mathematics of computation, 44(170), pp.519-521.
- Menezes, A.J., Van Oorschot, P.C. and Vanstone, S.A., 2018. Handbook of applied cryptography. CRC press.
- Schindler, W., 2000, August. A timing attack against RSA with the chinese remainder theorem. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 109-124). Springer, Berlin, Heidelberg.
- Coppersmith, D., 1997. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. Journal of cryptology, 10(4), pp.233-260.

Appendix

Appendix - Variance estimation in Kocher's attack

The quantity $\mathrm{Var}(T - T')$ can be estimated with the formula

$$\frac{1}{s-1}\sum_{j=1}^{s}\left((T_j - T'_j) - \frac{1}{s}\sum_{j=1}^{s}(T_j - T'_j) \right)^2,$$

recalling that:

- s is the number of generated ciphertexts;
- T_j is the sum of the time required to decrypt the j-th ciphertext;
- $T'_j = e + \sum_{i=0}^{b-1} t'_i, \qquad \forall j \in \{1, \dots, s\}$;
- e is the overhead time required by the attacker custom decryption;
- $b-1$ is the index of the guessed bit.

Probability of a correct guess in Kocher's attack (1/5)

An attacker is performing the b-th iteration of the Kocher's attack.
We can estimate the probability that d_b^* is correct.
Suppose the attacker:

- knows the first $b-1$ bits of the private exponent x;
- can measure $T' = \sum_{i=0}^{b-1} t'_i$ for each ciphertext y_j, with $j \in \{1, \dots, s\}$;

If x_b is correct, $T - T'$ yields $e + \sum_{i=0}^{k-1} t_i - \sum_{i=0}^{b-1} t_i = e + \sum_{i=b}^{k-1} t_i$.

Probability of a correct guess in Kocher's attack (2/5)

Now, assume that:

- all the time measurements i.i.d. as $\mathcal{N}(0,1)$;
- $\text{Var}(T_j - T'_j) = \text{Var}(e + \sum_{i=b}^{k-1} t_i)$ for each ciphertext y_j;
- the expected variance among all ciphertexts: $\text{Var}(e) + (k-b)\nu$, with $\nu := \text{Var}(t_i)\ \forall i$.

However, if only the first $c < b$ bits of the exponent guess are correct, the expected variance will be $\text{Var}(e) + (k + b - 2c)\nu$.

Probability of a correct guess in Kocher's attack (3/5)

Finally, assuming Var(e) negligible, we can state that the following two probabilities are the same:

1. that subtracting a correct t'_b from each ciphertext will reduce the total variance more than subtracting an incorrect t'_b;
2. that d^*_b is correct.

In the next two slides, we will show formulas to attain the first probability.

Probability of a correct guess in Kocher's attack (4/5)

$$
\begin{aligned}
&Pr\left[\frac{1}{s-1}\sum_{j=1}^{s}\left(\sqrt{k-b}X_j+\sqrt{2(b-c)}Y_j-0\right)^2>\frac{1}{s-1}\sum_{j=1}^{s}\left(\sqrt{k-b}X_j-0\right)^2\right]\\
&=Pr\left[(k-b)\sum_{j=1}^{s}X_j^2+2(b-c)\sum_{j=1}^{s}Y_j^2+\sqrt{2(b-c)(k-b)}\sum_{j=1}^{s}X_jY_j>(k-b)\sum_{j=1}^{s}X_j^2\right]\\
&=Pr\left[2(b-c)\sum_{j=1}^{s}Y_j^2+\sqrt{2(b-c)}\sqrt{k-b}\sum_{j=1}^{s}X_jY_j>0\right]\\
&=Pr\left[2\sqrt{2(b-c)(k-b)}\sum_{j=1}^{s}X_jY_j+2(b-c)\sum_{j=1}^{s}Y_j^2>0\right]
\end{aligned}
$$

where $X\sim\mathcal{N}(0,1)$ and $Y\sim\mathcal{N}(0,1)$.

Probability of a correct guess in Kocher's attack (5/5)

Moreover, for s large enough:

- $\sum_{j=1}^{s} Y_j^2 \approx s$
- $\sum_{j=1}^{s} X_j Y_j \sim \mathcal{N}(0, \sqrt{s})$,

yielding

$$
\begin{aligned}
Pr\left(2\sqrt{2(b-c)(k-b)}\left(\sqrt{s}Z\right) + 2(b-c)s > 0\right) &= Pr\left(Z > -\frac{\sqrt{s(b-c)}}{2(k-b)}\right) \\
&= Pr\left(Z < \frac{\sqrt{s(b-c)}}{2(k-b)}\right) \\
&= \Phi\left(\sqrt{\frac{s(b-c)}{2(k-b)}}\right)
\end{aligned}
$$

where $\Phi(x)$ is the cumulative density function of $\mathcal{N}(0,1)$.