A template attack against Verify PIN algorithms

Hélène Le Bouder, Thierno Barry, Damien Couroussé,
Jean-Louis Lanet and Ronan Lashermes

Personal Identification Number (PIN) codes.

- Used to authenticate the user,
- in payment cards or SIM cards...
- Targets of choice for malicious adversaries.
- A limited number of trials.

Personal Identification Number (PIN) codes.

- Used to authenticate the user,
- in payment cards or SIM cards...
- Targets of choice for malicious adversaries.
- A limited number of trials.

Side Channel Analysis (SCA)

- SCA consists in observing some physical characteristics which are modified during the computation performed on the circuit.
- Most classic leakages are: timing, power consumption, electromagnetic emissions (EM) ...
- The main difficulty of the attack is to succeed with very few traces.
- Template attack is a kind of SCA, based on characterization.

Verify PIN algorithm

```
procedure VERIFY PIN(candidate PIN V)
    counter = counter − 1
    if counter > 0 then
        status = COMPARISON(U,V)
        status2 = COMPARISON(U,V)
        if status ≠ status2 then
            ERROR, device is blocked
        else
            if status = TRUE then
                counter initialized at original value.
            end if
        end if
    else
        device is blocked
    end if
    return status
end procedure
```

- PIN code is an array of m bytes.
- **True PIN**: U,
- **Candidate PIN**: V,
- $U \in [\![0, 9]\!]^m$.
- 10^m different PIN codes.
- Countermeasure against fault attack: compare U and V twice.

Comparison of two PIN codes

```
procedure COMPARISON(candidate PIN V, true PIN U)
    status = FALSE
    diff = FALSE
    fake = FALSE
    for b = 0 to m do
        if U_b ≠ V_b then
            diff = TRUE
        else
            fake = TRUE
        end if
        if (b = m) and (diff = FALSE) then
            status = TRUE
        else
            fake = TRUE
        end if
    end for
    return status
end procedure
```

Countermeasure against timing attack:
comparison between U and V has to be in a constant time.

A template attack

2 phases

1. profiling phase,
2. attack phase.

The attacker can :

- obtain one trace on the targeted device;
- change the True PIN in her profiling device;
- obtain many traces on her profiling device.

On the profiling device

Step 1: Campaign on the profiling device

- Campaign is for one given byte b.
- The byte U_b of the True PIN takes all values k in $[\![0, 9]\!]$ and the other bytes stay to zero.
- Bytes of Candidate PIN V are fixed to a chosen value v.
- For each (k, v) collect many traces: $M_{v,k} = \left\{xk_{(i,j)}\right\}$, i for trace, j for time.

On the profiling device

Step 1: Campaign on the profiling device

- Campaign is for one given byte b.
- The byte U_b of the True PIN takes all values k in $[\![0, 9]\!]$ and the other bytes stay to zero.
- Bytes of Candidate PIN V are fixed to a chosen value v.
- For each (k, v) collect many traces: $M_{v,k} = \left\{xk_{(i,j)}\right\}$, i for trace, j for time.

Step 2: Detection of points of interest.

Select the moment of computation of Comparison (relevant j).

On the profiling device

Step 1: Campaign on the profiling device

- Campaign is for one given byte b.
- The byte U_b of the True PIN takes all values k in $[\![0, 9]\!]$ and the other bytes stay to zero.
- Bytes of Candidate PIN V are fixed to a chosen value v.
- For each (k, v) collect many traces: $M_{v,k} = \left\{xk_{(i,j)}\right\}$, i for trace, j for time.

Step 2: Detection of points of interest.

Select the moment of computation of Comparison (relevant j).

Step 3: Build of templates.

- Compute the covariance matrix $S_{v,k} = \{sk_{(j,j')}\}$,
  $sk_{(j,j')} = \frac{1}{n-1} \cdot \left(xk_j - \overline{xk_j}\right)^t \left(xk_{j'} - \overline{xk_{j'}}\right)$.

On targeted device

Step 4: Campaign on the targeted device

- True PIN byte U_b is **unknown**, it is the target;
- Candidate PIN byte V_b is equal to v.
- Trace is a vector $T_v = \{x_j\}$.

On targeted device

Step 4: Campaign on the targeted device

- True PIN byte U_b is **unknown**, it is the target;
- Candidate PIN byte V_b is equal to v.
- Trace is a vector $T_v = \{x_j\}$.

Step 5: Confrontation between measurements

- Confront the trace T_v to the template matrix $S_{v,k}$.
- General formula in template attack:
$$F_v\left(T_v|S_{v,k},\overline{xk}\right) = \frac{1}{\sqrt{2\pi^P \cdot |S_{v,k}|}} \cdot \exp\left(-\frac{1}{2}\cdot\left(T_v - \overline{xk}\right)\cdot S_{v,k}^{-1}\cdot\left(T_v - \overline{xk}\right)^t\right).$$

On targeted device

Step 4: Campaign on the targeted device

- True PIN byte U_b is **unknown**, it is the target;
- Candidate PIN byte V_b is equal to v.
- Trace is a vector $T_v = \{x_j\}$.

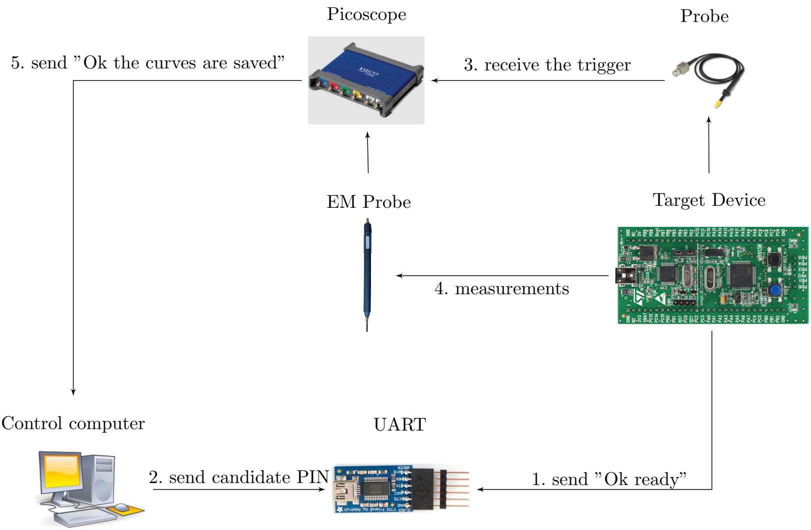
Step 5: Confrontation between measurements

- Confront the trace T_v to the template matrix $S_{v,k}$.
- General formula in template attack:
$F_v\left(T_v | S_{v,k}, \overline{xk}\right) = \frac{1}{\sqrt{2\pi^P \cdot |S_{v,k}|}} \cdot \exp\left(-\frac{1}{2} \cdot \left(T_v - \overline{xk}\right) \cdot S_{v,k}^{-1} \cdot \left(T_v - \overline{xk}\right)^t\right).$

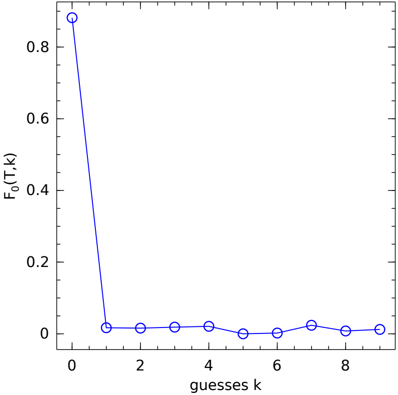
Step 6: Discriminating guesses

- Return the guess k_v for which F_v is maximal for a given T_v.
- Rank the guesses k according to the value of $F_v(T_v, k)$.

Test bench

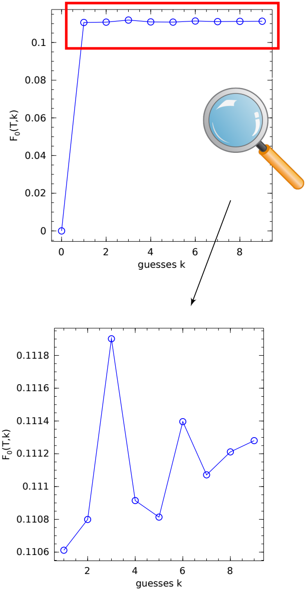

Picoscope
Probe
5. send "Ok the curves are saved"
3. receive the trigger
EM Probe
Target Device
4. measurements
Control computer
UART
2. send candidate PIN
1. send "Ok ready"

General results



- The True byte PIN:
  $U_b = 0$
- The Candidate byte PIN:
  $V_b = 0$
- The returned guess is clearly:
  $k = 0$
- If $U_b = V_b$.
  The attack always succeeds.

General results



- The True PIN byte: $U_b = 3$.
- The Candidate PIN byte: $V_b = 0$.
- The returned guess is $k = 3$.
- $U_b \neq V_b$: The attack succeeds, not so clearly.

Final attack

procedure ATTACK(C the number of trials in the VERFY PIN)
  $N = C - 1$ // limitation of number trials.
  $v = 0$
  $\mathbb{K} = [\![0, 9]\!]$
  $\hat{k} = \max^{-1}_{k \in \mathbb{K}} (F_v(T_v, k))$ // $\hat{k}$ best guess with v.
  while $\hat{k} \neq v$ and $N > 0$ **do**
    $N = N - 1$
    $\mathbb{K} = \mathbb{K} \setminus \{v\}$ // guess v is eliminated.
    $v = \hat{k}$
    $\hat{k} = \max^{-1}_{k \in \mathbb{K}} (F_v(T_v, k))$.
  end while
  return $\hat{k}$
end procedure

- v is the value tested on the Candidate PIN: $V_b = v$.
- $F_v(T_v, k)$ function template of the attack.

1. Send candidate PIN with all bytes to 0.
2. Then test the PIN code returned by the first attack.

- **Worst case**: in 8 trials, the PIN code is retrieved.

Final attack

Success rate

number of traces:		1	2	3	4	5	6	7	8
$n = 100000$	1 COMPARAISON	27.70	41.47	53.84	63.99	73.07	81.33	88.51	100
	2 COMPARAISON	31.71	46.56	57.82	67.76	76.63	84.36	90.68	100
$n = 200000$	1 COMPARAISON	29.28	44.27	56.79	67.41	76.66	83.91	90.68	100
	2 COMPARAISON	32.72	49.52	61.96	72.05	80.49	87.53	93.23	100
$n = 400000$	1 COMPARAISON	29.56	44.11	56.0	66.88	75.96	84.04	90.58	100
	2 COMPARAISON	32.91	48.38	60.88	71.68	80.07	86.91	92.94	100

Success rate to retrieve a byte of a True PIN U_b according to the size n of the templates and the number and the choice of traces.

- The first SCA attack with EM traces on Verify PIN algorithms.
- To enter a PIN code, a user has a limited number of trials.
- Therefore the main difficulty of the attack is to succeed with very few traces.
- The PIN is retrieved in 8 trials at most!

- The first SCA attack with EM traces on Verify PIN algorithms.
- To enter a PIN code, a user has a limited number of trials.
- Therefore the main difficulty of the attack is to succeed with very few traces.
- The PIN is retrieved in 8 trials at most!
- It becomes a new real threat, and it is feasible on a low cost and portable platform.
- Some protections against fault attacks introduce new vulnerabilities.

- The first SCA attack with EM traces on Verify PIN algorithms.
- To enter a PIN code, a user has a limited number of trials.
- Therefore the main difficulty of the attack is to succeed with very few traces.
- The PIN is retrieved in 8 trials at most!
- It becomes a new real threat, and it is feasible on a low cost and portable platform.
- Some protections against fault attacks introduce new vulnerabilities.
- **Future works:**
  - Find new contermeasures.
  - Test the attack on a real device (mobile phone or smart card).

Thank you for your attention !

Do you have any questions?