SDR as Side Channel Attack Platform

Jan Ruge (bolek42)

Who Am I

- I'm bolek42
- Bachelor at Univeristy Hamburg
- Now Master at TU-Darmstadt

Side-Channel Attacks

- Uses physical characteristics of an implementation
  - Power Consumption
  - Timing behavior
  - Electromagnetic emmanation (this work)
  - Can be used to detect changes in programm flow

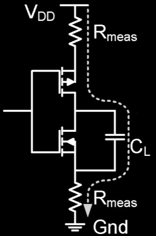
Why SDR for Sidechannel Attacks

- Digital Oscilloscope are expensive
  - Often limited memory depth
- Cheap SDRs are available (e.g. RTL-SDR)
  - Many DSP frameworks available (e.g. GNURadio)
  - Analog filtering and amplification
  - Fast Analog-Digital Converter

Electromagnetic Emmanation

Origin of Side-Channel Effects

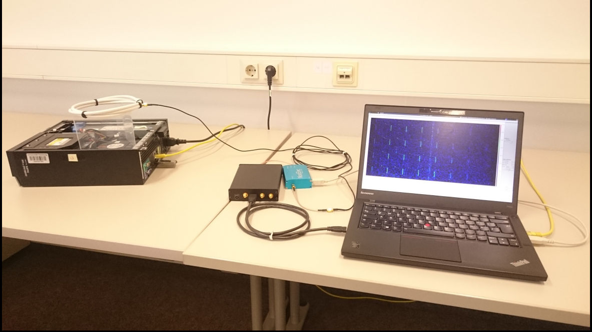
- The Power consumption of the CPU is changing with Operations
- Hamming Distance Model
  - The power consumption is correlating with the number of flipping Bits



- CPU Voltage Regulation
  - Compensates the fluctuating power consumption
  - Creates low frequency noise (< 4 MHz)

Example Setup for Desktop PC

- Attack based on the work by Genkin et. Al.
- Test Setup:



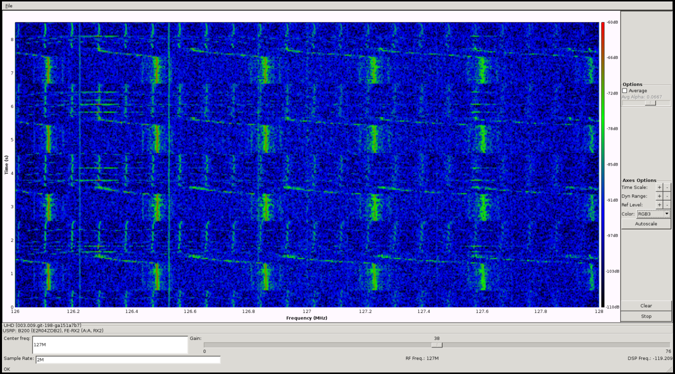
- Side-Channel effects located at < 4 MHz
- Bandwidth 100 kHz (RTL-SDR or rad1o can be used)
- But an upconverter (e.g. Ham It Up) required

First Experiment

- Test Program

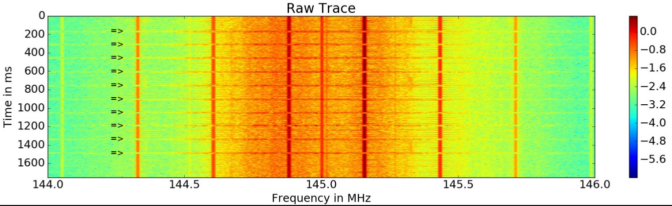
```
while 1:
    for i in xrange(40000000): pass
    time.sleep(1)
```

- Raw Spectrogram

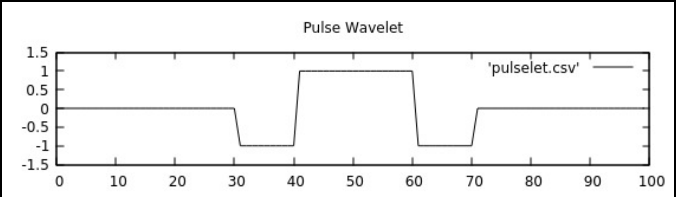


Finding Trigger Frequencies

- Issue multiple challenges
- Challenges are visible as interruption of carrier

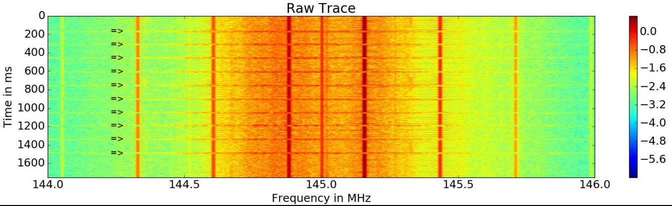


- Use multiple Pulse Wavelets

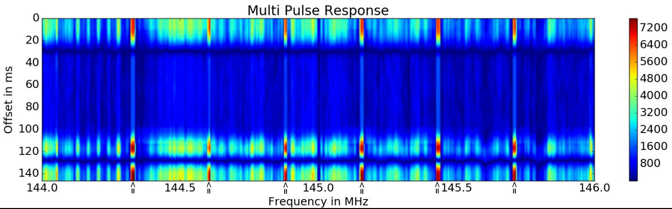


Finding Trigger Frequencies

- Raw Trace

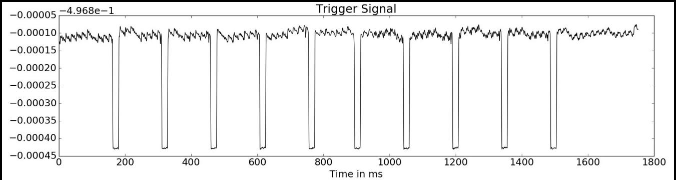


- Wavelet Response

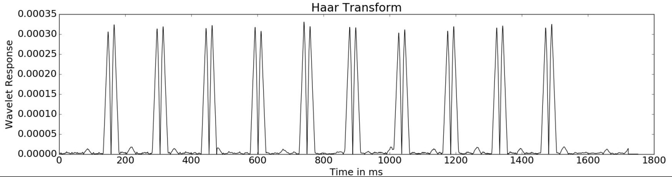


Extracting Traces

- Trigger frequency can be filtered by a GNURadio Flowgraph
- Amplitude demodulated trigger frequency



- Haar Wavelet Response (Slope Detection)



- In addition Static alignment is used for better results

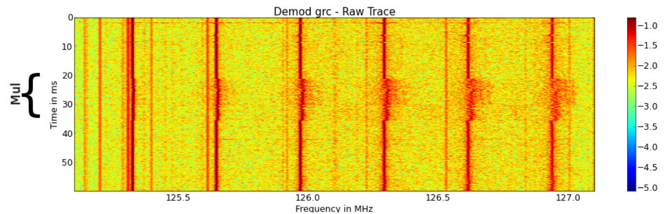
OpenSSL Multiplication

Example for Side Channel Effects

- Test Program for OpenSSL

```
for (i=0; i < 8000000; i++) i ^= 0;
for (i=0; i < 400; i++) BN_mod_mul(r,r,arg,N);
for (i=0; i < 8000000; i++) i ^= 0;
```

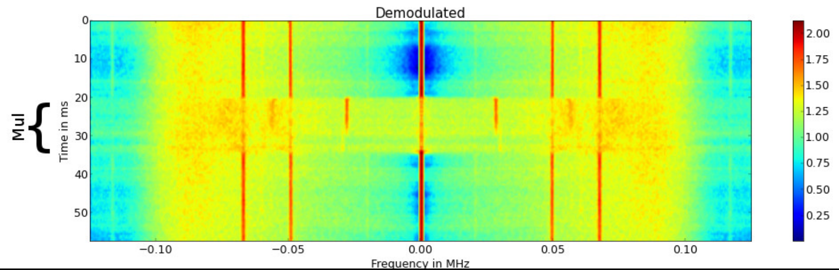
- Raw Spectrogram



- Looks like Frequency Modulation?

Demodulation

- Use GNURadio to isolate carrier
- Frequency demodulated carrier (averaged):



- Multiplications are clearly visible
- In some cases demodulation is not required

Differential Power Analysis by Paul Kocher

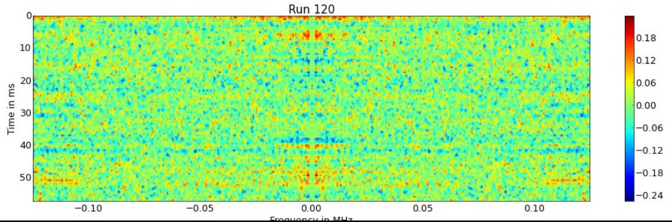
- Used to find differences in Side-Channel Effects
  - Choose two arguments A and B
  - Perform multiple measurements with A and B
  - Compute $dpa = E(A) - E(B)$

1. $dpa \to 0$
   - A and B causing <u>same</u> Side-Channel effects
2. $|dpa| > 0$
   - A and B causing <u>different</u> Side-Channel effects

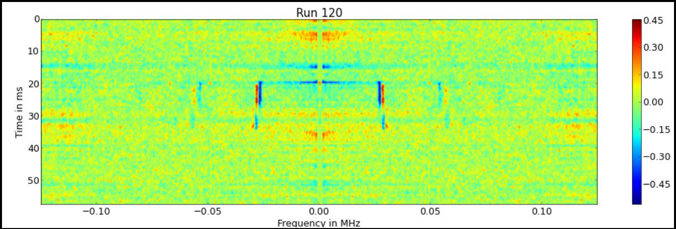
- Will be directly used on Spectrograms

Results OpenSSL Multiplication

- A < N, B < N:



- A < N, B > N:



OpenSSL Exponentiation

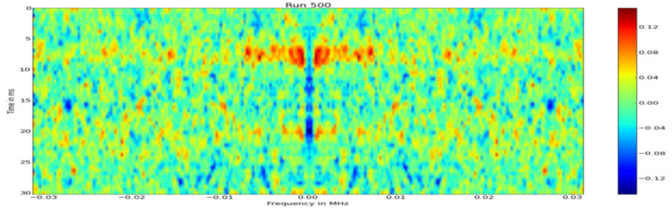
m-Array Exponentiation

- OpenSSLs Exponentiation Routine

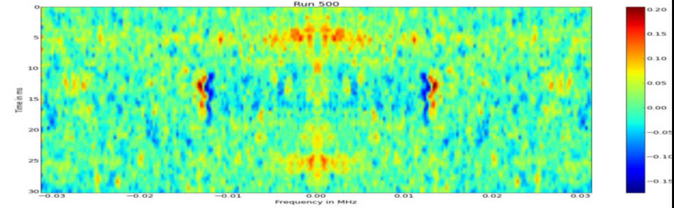
```
function m_array_exp(c,d,N) // c^d mod N
    c = c mod N

    //pow[i] = c^i mod N
    pow[0] = 1
    for i = 1...m
        pow[i] = pow[i-1] * c mod N

    D = Fragmentation of d in m-Bit words
    k = length(D)

    r = D[k-1]
    for i = k-2...0
        res = r ^ (2^m) mod N
        if D[i] > 0
            r = r * pow[D[i]] mod N //i=1: SCE!

    return r
```

Results OpenSSL Exponentiation

- A < N, B < N:



- A < N, B > N:



Application to RSA

RSA (Rivest, Shamir und Adleman)

- Public Key Cryptosystem
- Key generation

$$N = pq$$
$$\varphi(N) = (p-1)(q-1)$$
$$e \in \mathbb{Z}_{\varphi(N)}$$
$$d \equiv e^{-1} \mod \varphi(N)$$

- Encryption with Public-Key: $c = m^e \mod N$
  - Public Key: (e, N)
- Decryption with Private-Key: $m = c^d \mod N$
  - Private Key: (d, N, p, q)
- Coppersmith: Knowledge of upper half of p breaks RSA

RSA with Chinese Remainder Theorem

- Regular RSA decryption: $m = c^d \mod N$
- RSA-CRT:

$$c_p = c^{d \mod (p-1)} \mod p$$
$$c_q = c^{d \mod (q-1)} \mod q$$
$$m = ((q^{-1} \mod p)(c_p - c_q) \mod p)q + c_q$$

- The modul for the exponentiation is p or q !
- Coppersmith: Knowledge of upper half of p breaks RSA

RSA Blinding (default in OpenSSL)

- Blinding message

$$c_b = c \cdot r^d \mod N$$

- Application of RSA

$$\begin{aligned} m_b &= c_b^e \mod N \\ &= c^e \cdot r^{ed} \mod N \\ &= m \cdot r \mod N \end{aligned}$$

- Unblinding message

$$\begin{aligned} m &= m_b \cdot r^{-1} \mod N \\ &= m \cdot r \cdot r^{-1} \mod N \\ &= m \end{aligned}$$

Binary Search on secret Prime

- In case of RSA-CRT the modul is p or q
- Assume c is not blinded
- Side-Channel can be used for binary search on CRT-Modul
- Attack pseudocode:

```
reference = 0b111110....
secret = 0
for i = (n-1)..0
    if DPA(secret + 2^i, reference) > x
        secret += 2^i //secret + (2^i) < p
    else
        continue      //secret + (2^i) > p

return secret
```

- Coppersmith: Knowledge of upper half of p breaks RSA

Demo Video

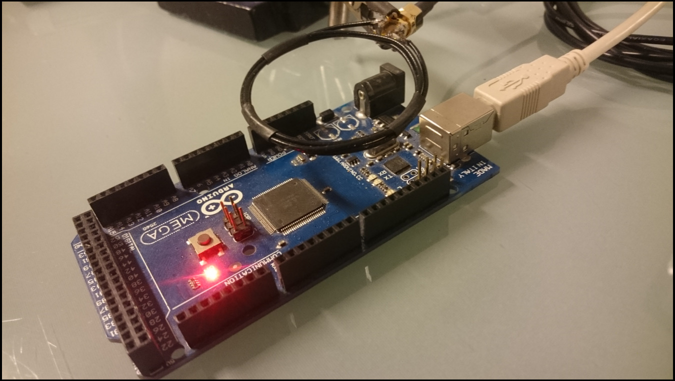
5h for 84 bits...

Slow but works

New Device - Arduino

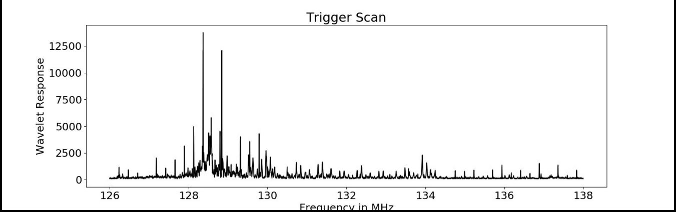
Now for Arduino

- Sidechannel Effects differ from device to device
- Emmanation from the Powersupply

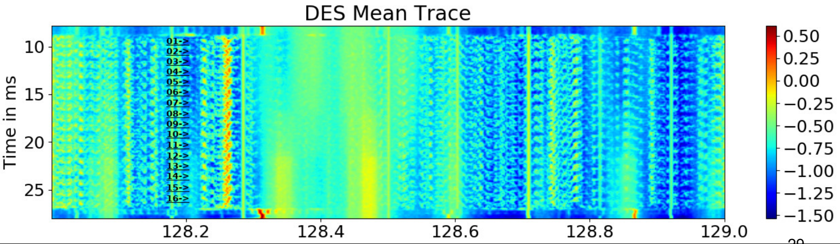


DES on Arduino (16MHz)

- Using Wavelet method to scan for Sidechannel Effects

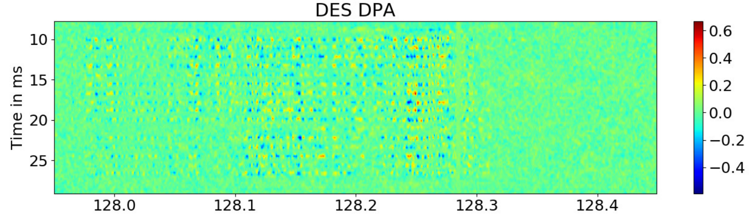


- Averaged Spectrogram of DES computation
- Individual rounds are distinguishable

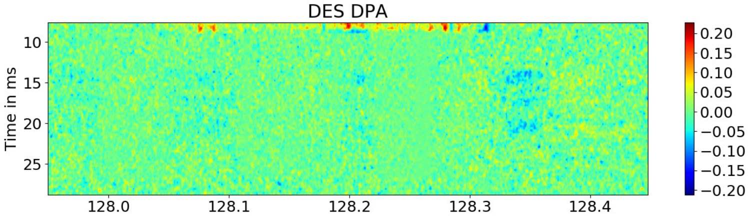


DES on Arduino - Data Dependent Leakage

- A=0000000000000000, B=ffffffffffffffff



- A=0000000000000000, B=0000000000000000



Conclusion

- GNURadio is awesome!
  - Create flowgraph with GUI
  - Import the compiled `top_block.py`
  - Profit!
- Desktop PC
  - Unprotected RSA is vulnerable
  - Very slow attack
- Arduino
  - Symmetric Crypto might be vulnerable
  - No key bits recovered yet :(
- Not tested
  - Mobile Devices?
  - Genkin et. Al. also did this
- Clone it, Hack it: github.com/bolek42/rsa-sdr

Thanks for your attention!

Q&A?