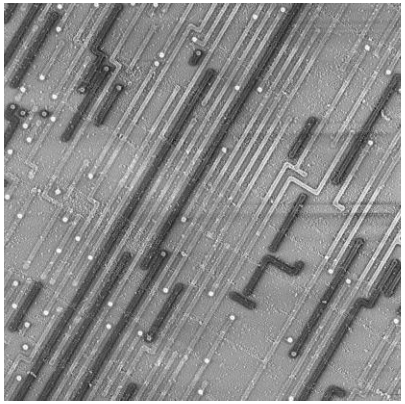
Efficient High-Speed WPA2 Brute Force Attacks using Scalable Low-Cost FPGA Clustering

Markus Kammerstetter, Markus Muellner, Daniel Burian[2], Christian Kudera and Wolfgang Kastner

Secure Systems Lab Vienna, Automation Systems Group,
Vienna University of Technology

Trustworks KG[2]

- Focus on Hardware Security and Physical Attacks
- Lab equipment - Trustworks KG
- Wide range of dedicated tools such as FIB, SEM, Plasma Etcher, Prober, Polisher, etc.



Our Research Fields

Embedded Software Security

Firmware Code Analysis
- Static Anaylsis
- Dynamic Analysis & Debugging

Firmware Fuzz Testing

Implementation Attacks

Side-Channel Attacks

Probing Attacks
- Firmware & Crypto Key Extraction

Fault Injection
- Firmware & Crypto Key Extraction

IC Reverse Engineering
- Algorithm Extraction
- Test Mode Security

High-Speed Cryptography

Efficient Code Breaking on FPGAs

Our Research Fields

Secure Systems Lab Vienna
Vienna University of Technology

Embedded Software Security

Firmware Code Analysis
  - Static Anaylsis
  - Dynamic Analysis & Debugging
Firmware Fuzz Testing

Implementation Attacks

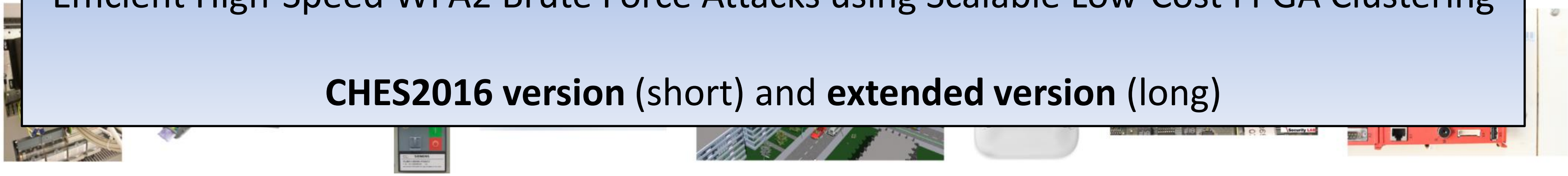
Side-Channel Attacks
Probing Attacks
  - Firmware & Crypto Key Extraction
Fault Injection
  - Firmware & Crypto Key Extraction
IC Reverse Engineering
  - Algorithm Extraction
  - Test Mode Security

High-Speed Cryptography

Efficient Code Breaking on FPGAs

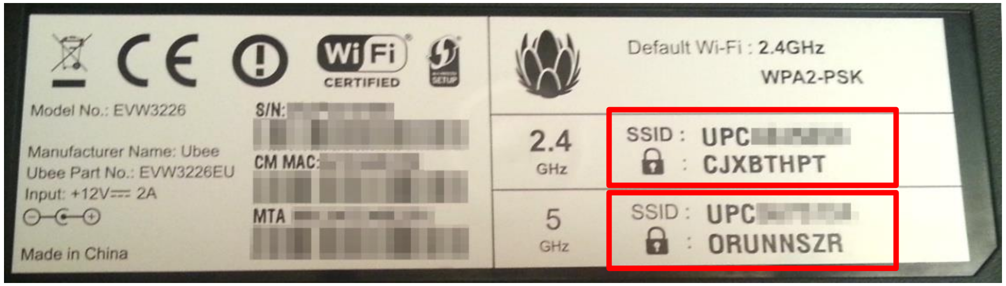
CHES2016: M. Kammerstetter, M. Muellner, D. Burian, C. Kudera and W. Kastner
Efficient High-Speed WPA2 Brute Force Attacks using Scalable Low-Cost FPGA Clustering

CHES2016 version (short) and **extended version** (long)



Problem Statement

- WPA2-Personal is omnipresent
- Minimum password length: 8 characters
- Embedded devices (routers, cable modems, …) frequently have bad default passwords
- Quality of password vs. cracking speed

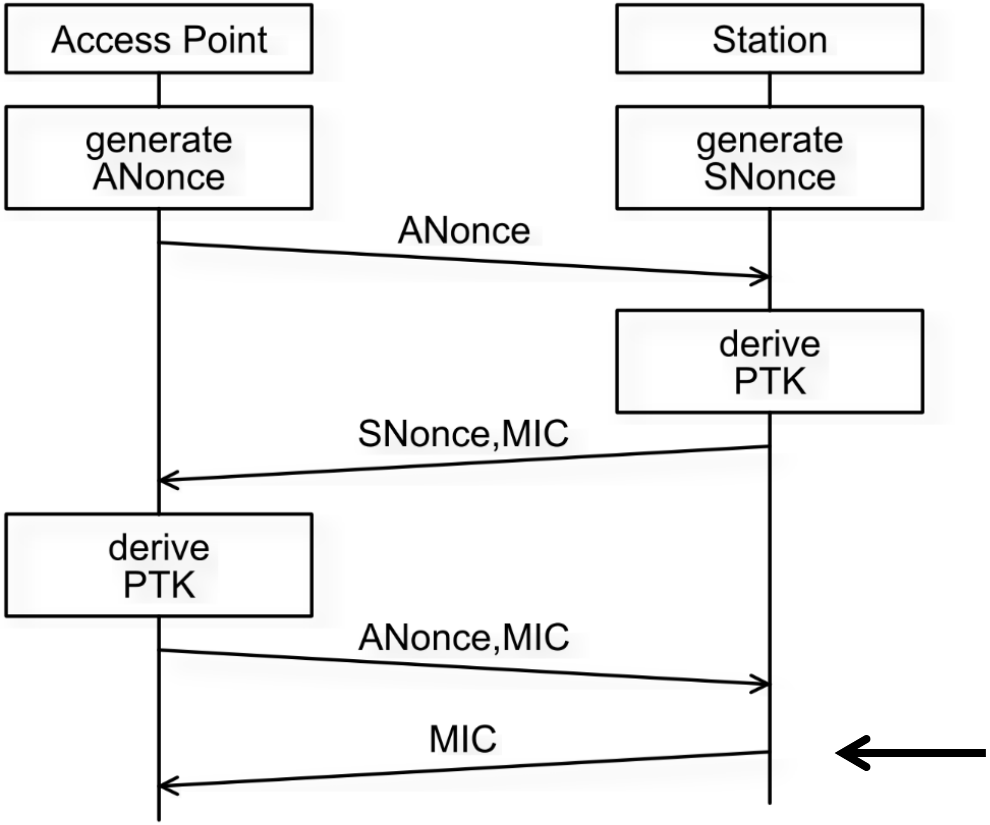


Model No.: EVW3226
S/N: [redacted]
Manufacturer Name: Ubee
Ubee Part No.: EVW3226EU
Input: +12V⎓ 2A
CM MAC: [redacted]
MTA [redacted]
Made in China

Default Wi-Fi : 2.4GHz
WPA2-PSK

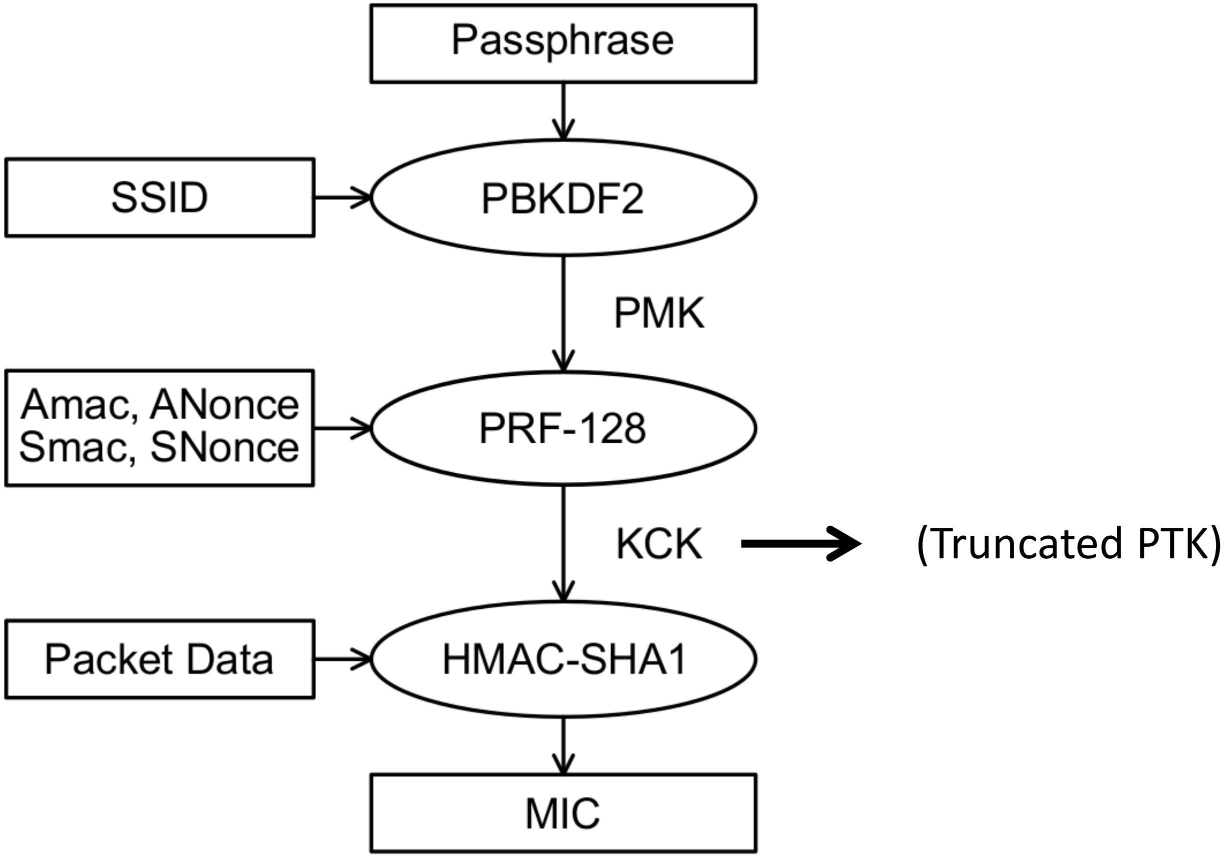
2.4 GHz	SSID : UPC[redacted] 🔒 : CJXBTHPT
5 GHz	SSID : UPC[redacted] 🔒 : ORUNNSZR

WPA2-Personal 4-Way Handshake



(MIC is computed over empty message with zero padding bytes **- known plaintext**)

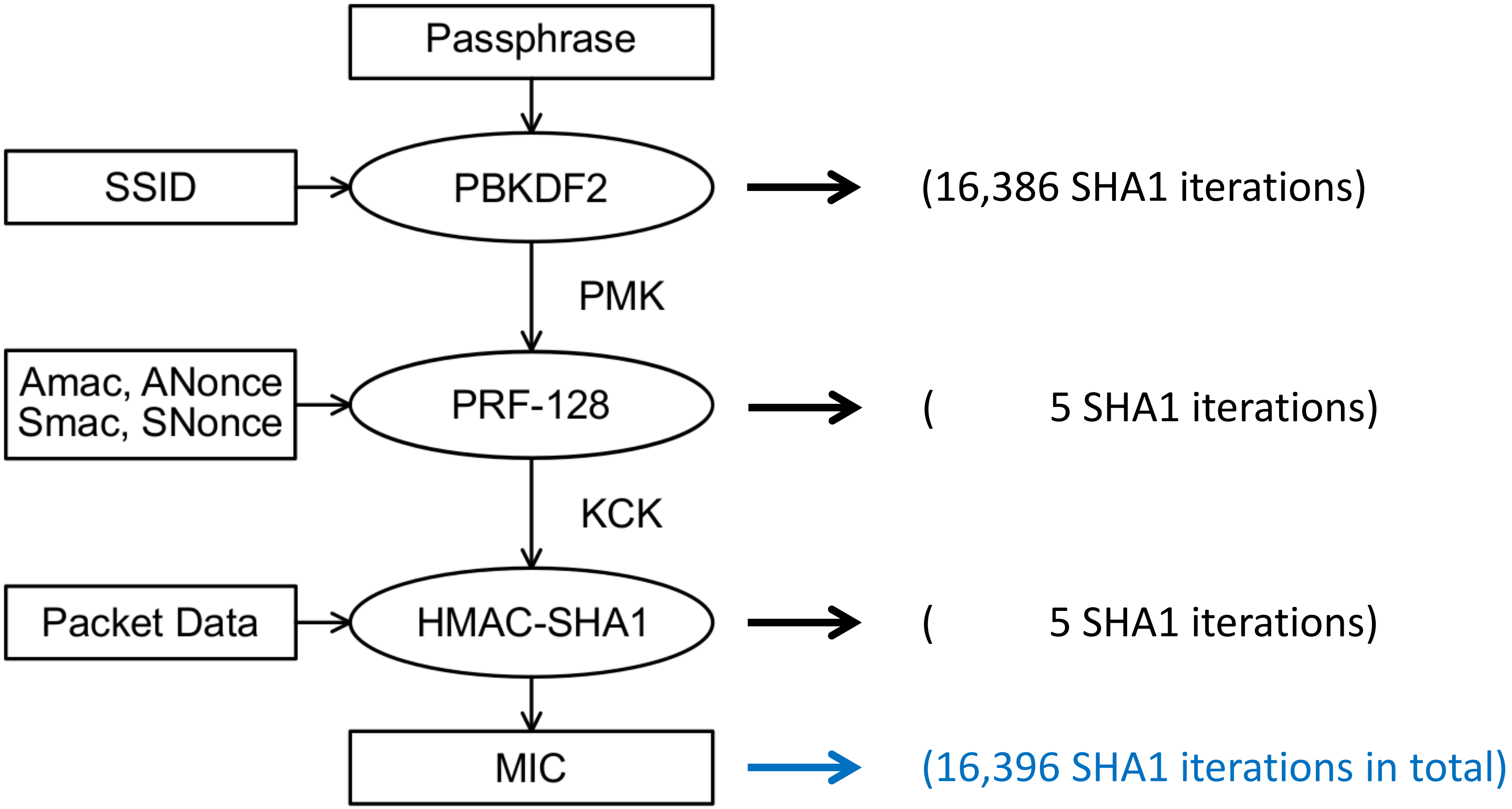
WPA2-Personal Key Derivation



Practical Attacks on Handshake

1. Attacker captures 4-way handshake (use of de-auth frames possible)
2. Choose password
3. Derive KCK (=truncated PTK) using password candidate and obtained SSID, MAC addresses and nonces
4. Password correct if computed MIC matches captured MIC

Computational Complexity



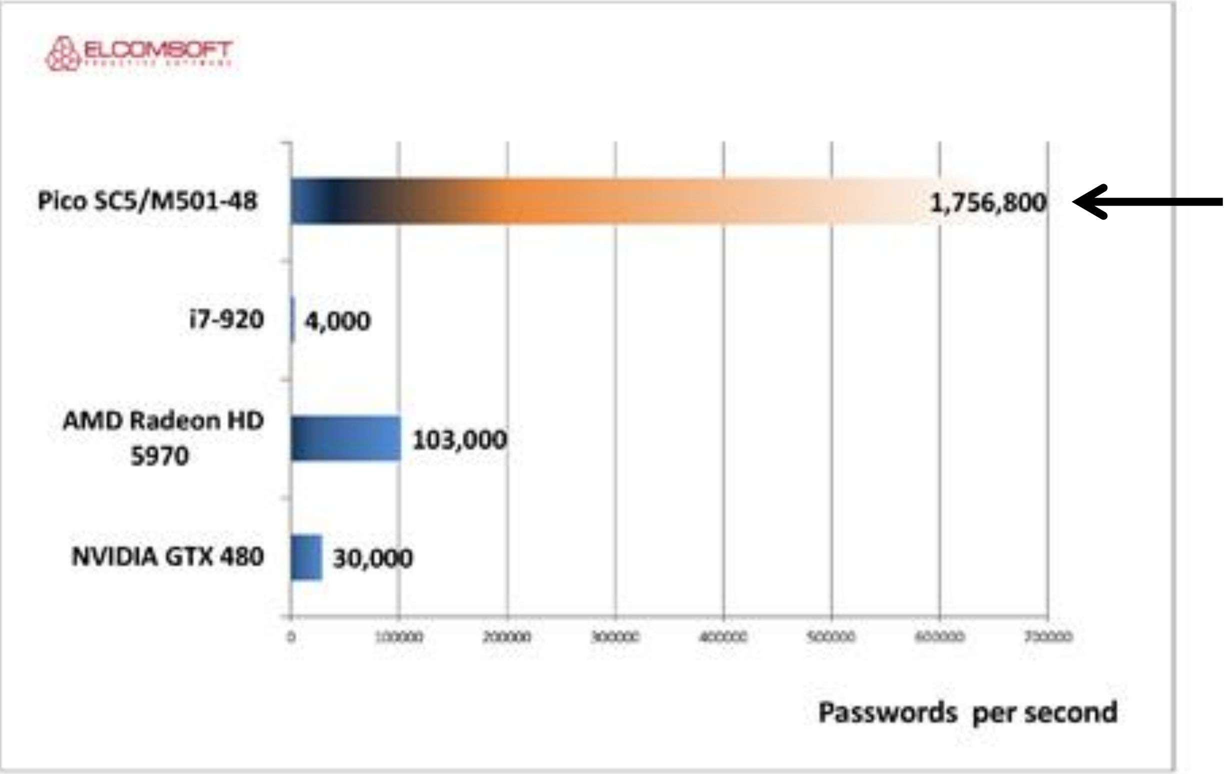
(16,386 SHA1 iterations)

(5 SHA1 iterations)

(5 SHA1 iterations)

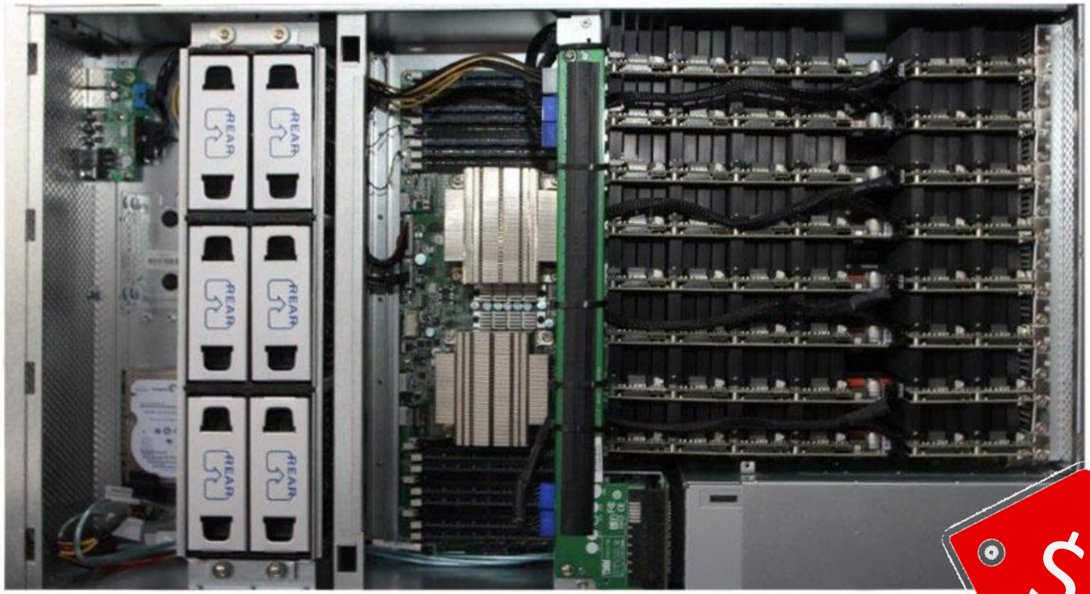
(16,396 SHA1 iterations in total)

How fast can we get ?



ElcomSoft and Pico Computing Demonstrate **World's Fastest** Password Cracking Solution
[https://www.elcomsoft.com/news/515.html]

Throwing Money at the Problem



1,988,360 keys/sec

48x Xilinx Kintex-7 K410T FPGA

[http://picocomputing.com/brochures/SC5-4U.pdf]

$128,000*

* Price request per e-mail to PicoComputing, April 22, 2015

Can we do better ?

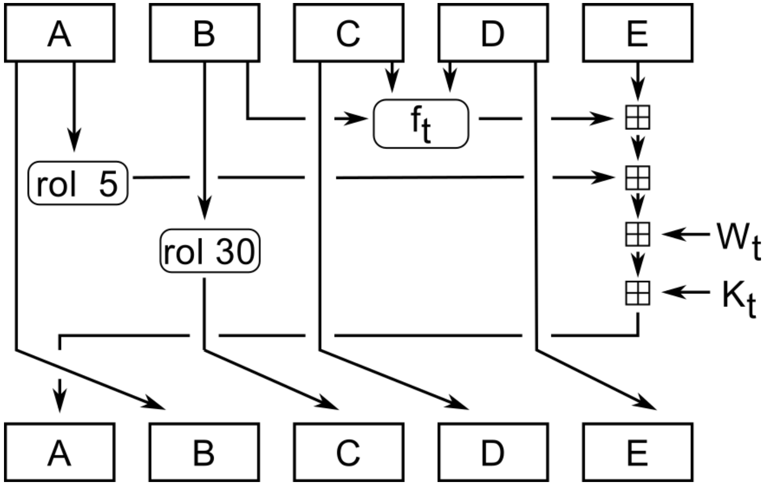
- SHA1 works on 512 bit chunks, 160 bit hash digest when finished
- 80 rounds (t)
- Message working schedule:

(Message broken up into 32bit chunks)

$$W_t = \begin{cases} M_t & 0 \leq t \leq 15 \\ rol(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}, 1) & 16 \leq t \leq 79 \end{cases}$$

Can we do better ?

- SHA1 rounds are based on compression:

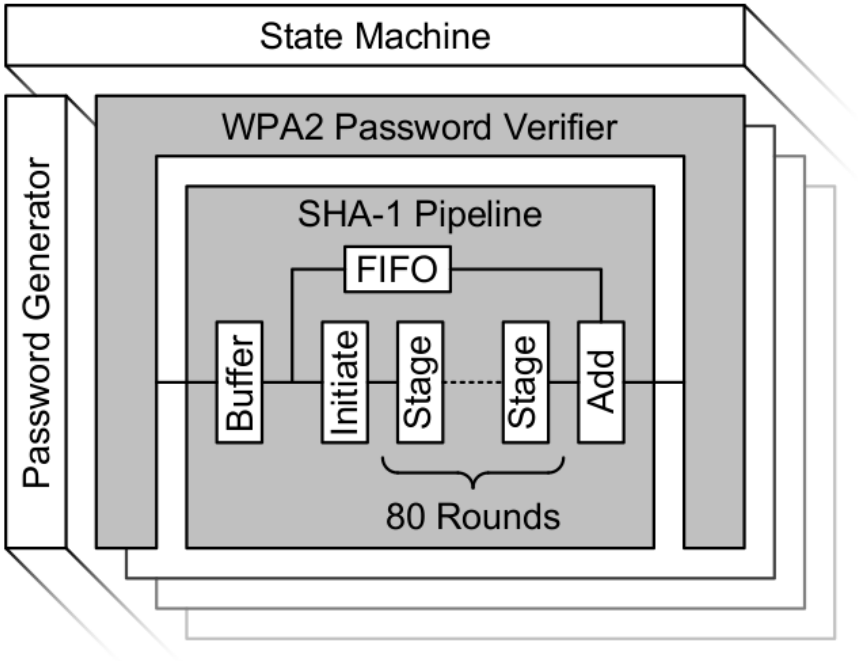


$$f_t = \begin{cases} (x \wedge y) \oplus (\neg x \wedge z) & 0 \leq t \leq 19 \\ x \oplus y \oplus z & 20 \leq t \leq 39 \\ (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) & 40 \leq t \leq 59 \\ x \oplus y \oplus z & 60 \leq t \leq 79 \end{cases}$$

$\boxplus$ → unsigned 32-bit addition

- Ideally suited for HW implementation, but addition is expensive (carry chain)

FPGA Design

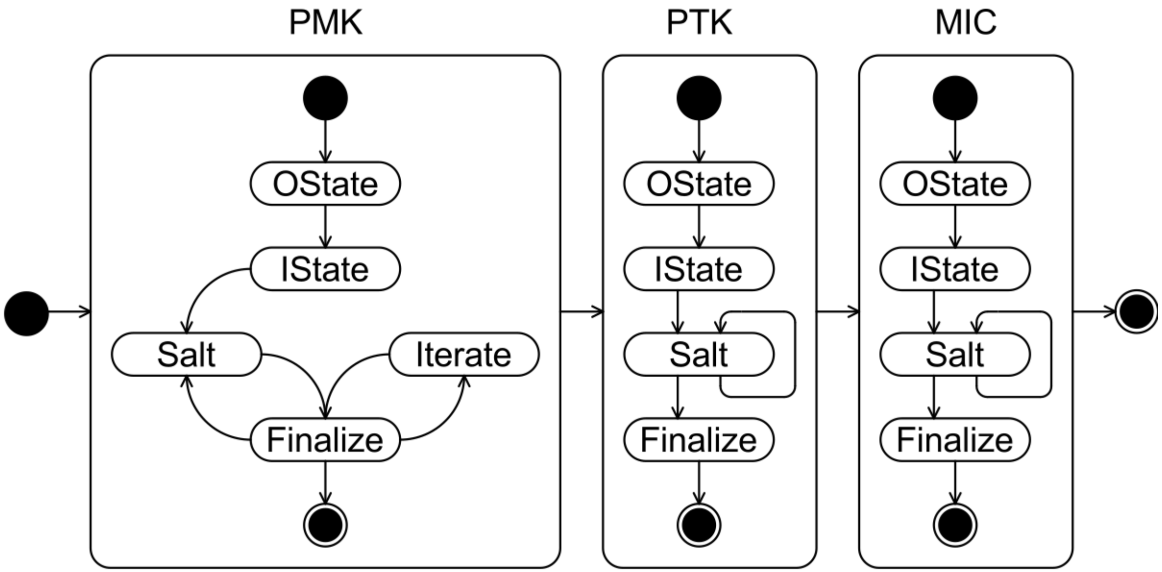


- Efficient SHA1 pipeline
- Goal: critical path delay reduction
- 83 stages (vs. 80 rounds)
- 3 additional stages:
  - *Buffer* stage (reduce pipeline input logic delay)
  - *Initiate* and *Add* stage splits up expensive addition of E word (carry chain delay)

SHA1 Pipeline Optimizations

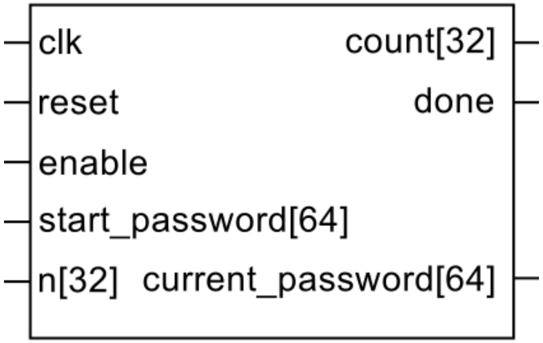
- Compute HMAC O-state first
(avoid storing intermediary result)
- Use of Block RAM delay lines instead of broad stage interconnects (avoid routing delays/congestion)
- State machine: many small multiplexers instead of a single big one
- Custom build parameters (e.g. for shift reg. inference)
- Extensive floor planning (explained later)

Key Derivation State Machine



Password Verifier

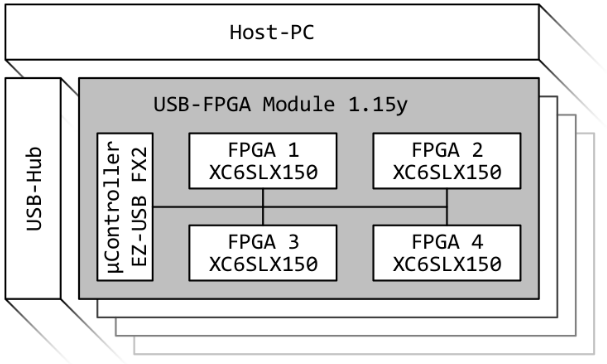
- Password generator realized as fast counter
- Verifier fills up all 83 stages of all cores



- Wait until computed MICs are available
- Compare computed MICs with captured MIC

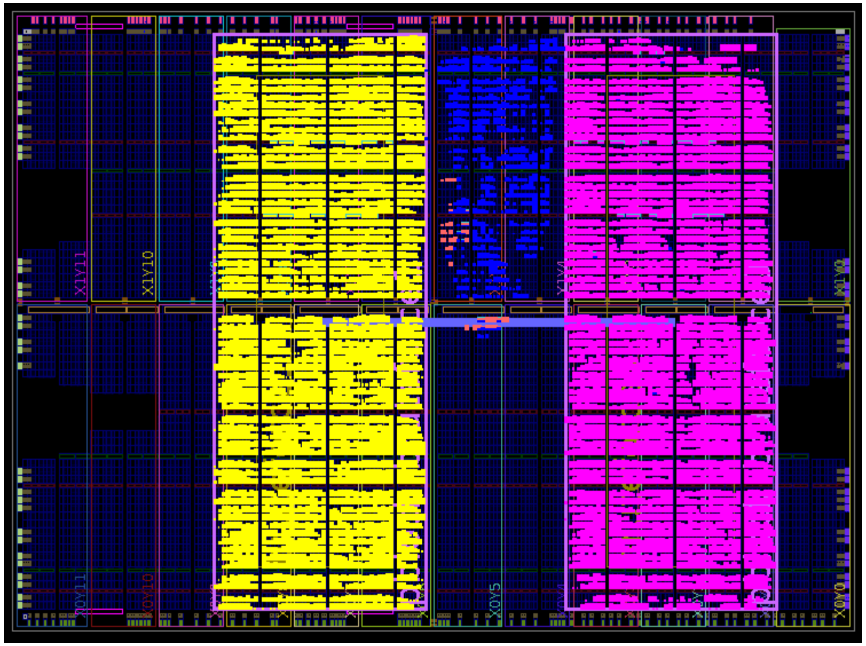
Implementations

- Focus on low-cost FPGAs
- Implementations for 3 different FPGAs:
  - Spartan 6 LX150T on Ztex 1.15y boards
  - Artix 7 200T on Ztex 2.16 board
  - Kintex 410T (for comparison purposes)



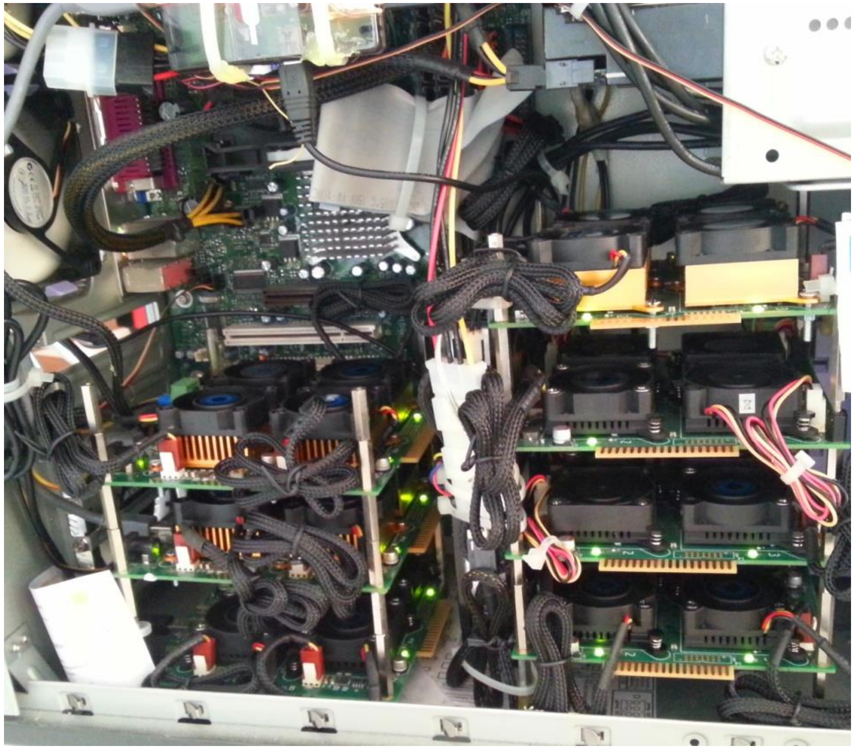
Spartan 6 LX 150T

- 2 Cores @ 180 MHz, dyn. frequency scaling based on error rate, 4 FPGAs per board



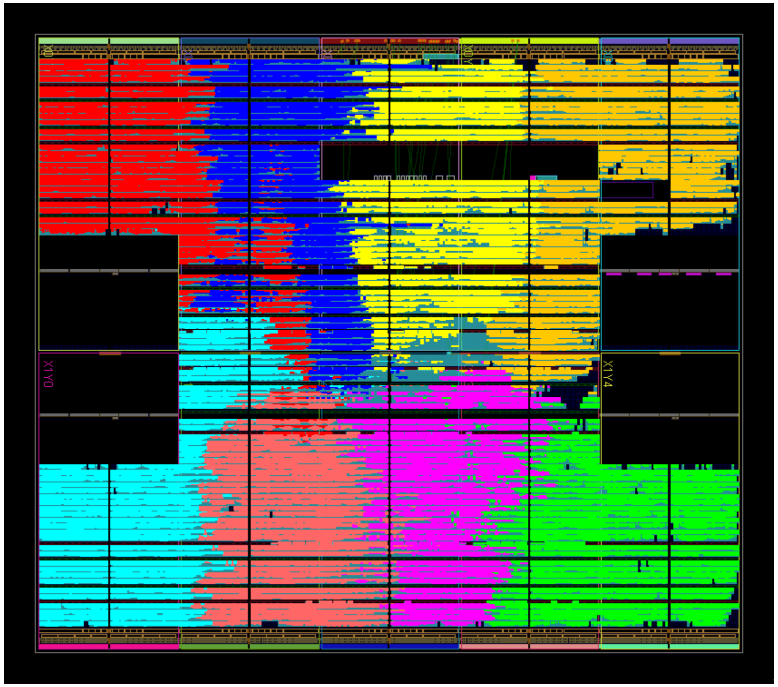
Spartan 6 LX160T Cluster

- 9 Ztex 1.15y boards, 4 FPGAs each (=36 FPGAs)
- 7 boards internal
- 2 boards external (development support)



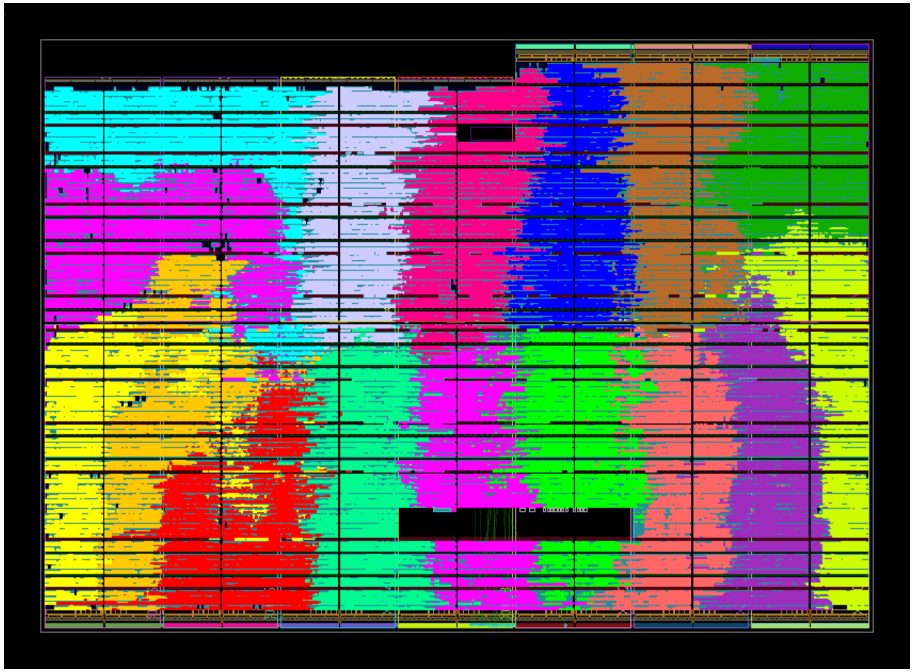
Artix 7 200T

8 Cores @ 180 MHz, dyn. Frequency scaling based on temperature, star topology



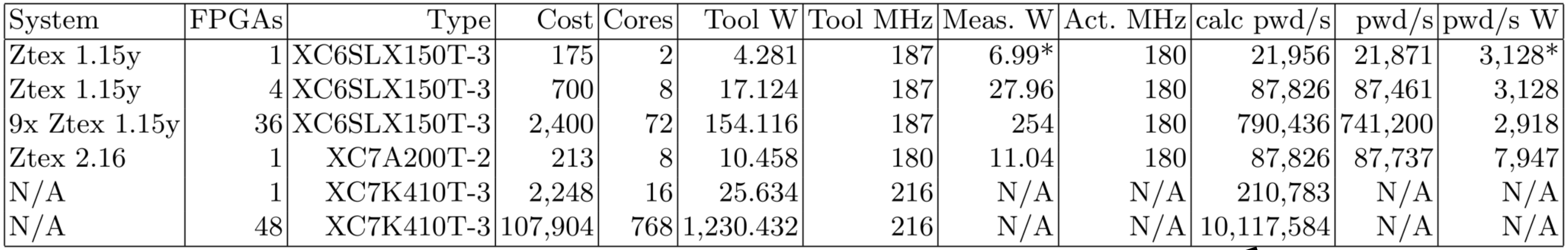
Kintex 410T

16 Cores @ 216 MHz, dyn. Frequency scaling based on temperature, star topolgy, simulations only



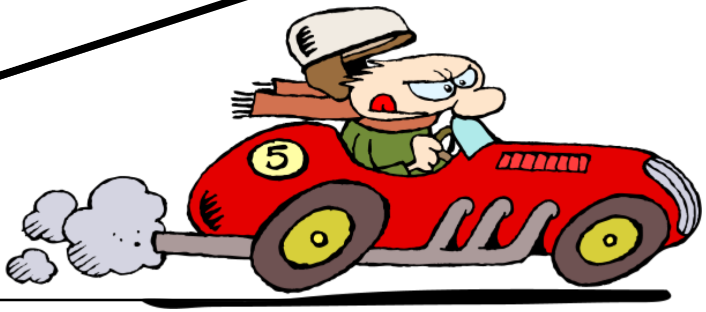
Performance & Power Consumption

Measured performance is close to calculated performance



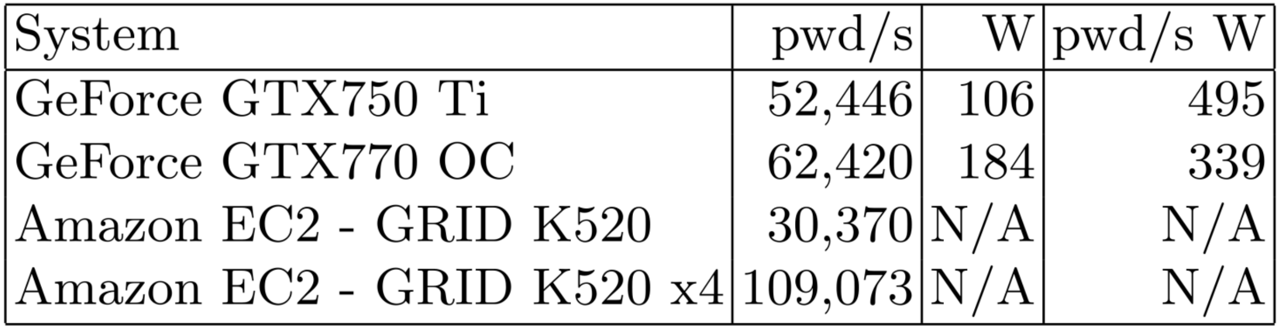
System	FPGAs	Type	Cost	Cores	Tool W	Tool MHz	Meas. W	Act. MHz	calc pwd/s	pwd/s	pwd/s W
Ztex 1.15y	1	XC6SLX150T-3	175	2	4.281	187	6.99*	180	21,956	21,871	3,128*
Ztex 1.15y	4	XC6SLX150T-3	700	8	17.124	187	27.96	180	87,826	87,461	3,128
9x Ztex 1.15y	36	XC6SLX150T-3	2,400	72	154.116	187	254	180	790,436	741,200	2,918
Ztex 2.16	1	XC7A200T-2	213	8	10.458	180	11.04	180	87,826	87,737	7,947
N/A	1	XC7K410T-3	2,248	16	25.634	216	N/A	N/A	210,783	N/A	N/A
N/A	48	XC7K410T-3	107,904	768	1,230.432	216	N/A	N/A	10,117,584	N/A	N/A

New speed record:
Compared to ElcomSoft's 1,988,360 keys/sec, that's 5.09x times as fast <u>on the same hardware</u>



GPU Comparison

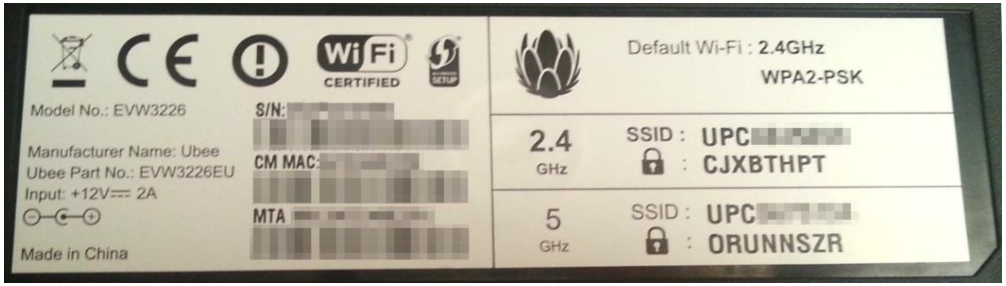
- Based on cudaHashcat v1.36
- GeForce GPUs in our lab machines
- GRID K250 GPUs on Amazon EC2 cloud



System	pwd/s	W	pwd/s W
GeForce GTX750 Ti	52,446	106	495
GeForce GTX770 OC	62,420	184	339
Amazon EC2 - GRID K520	30,370	N/A	N/A
Amazon EC2 - GRID K520 x4	109,073	N/A	N/A

Real World Case Study [extended version]

- UPC cable modems have weak default PW (8 characters, uppercase, [A..Z])



WiFi CERTIFIED

Model No.: EVW3226
S/N: [redacted]

Manufacturer Name: Ubee
Ubee Part No.: EVW3226EU
Input: +12V⎓ 2A
CM MAC: [redacted]
MTA [redacted]

Made in China

Default Wi-Fi : 2.4GHz
WPA2-PSK

2.4 GHz
SSID : UPC[redacted]
🔒 : CJXBTHPT

5 GHz
SSID : UPC[redacted]
🔒 : ORUNNSZR

- Assumption: If people change the password, it is likely that they also change the tedious SSIDs (i.e. UPC012345) to something meaningful

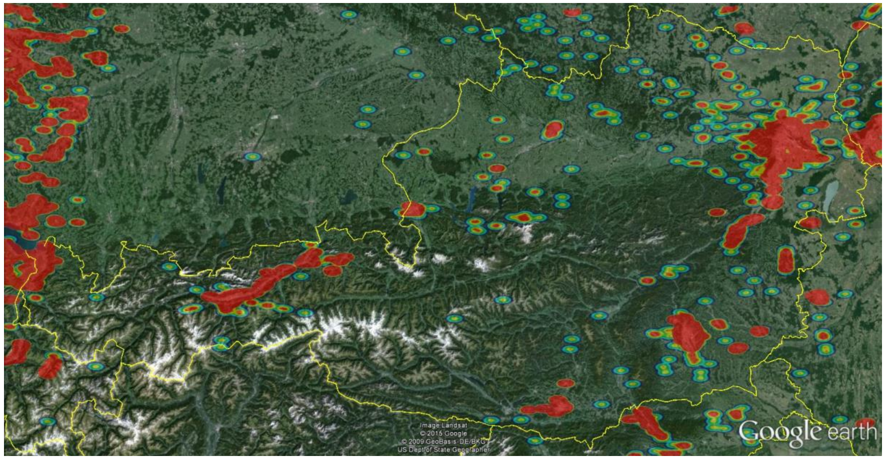
Real World Case Study [extended version]

- So we collected some handshakes …
  (own cable modems)
- Result: With our cluster (790,436 keys/sec) we can break the password in **3 days at most** !
- … but what's the real world impact ?

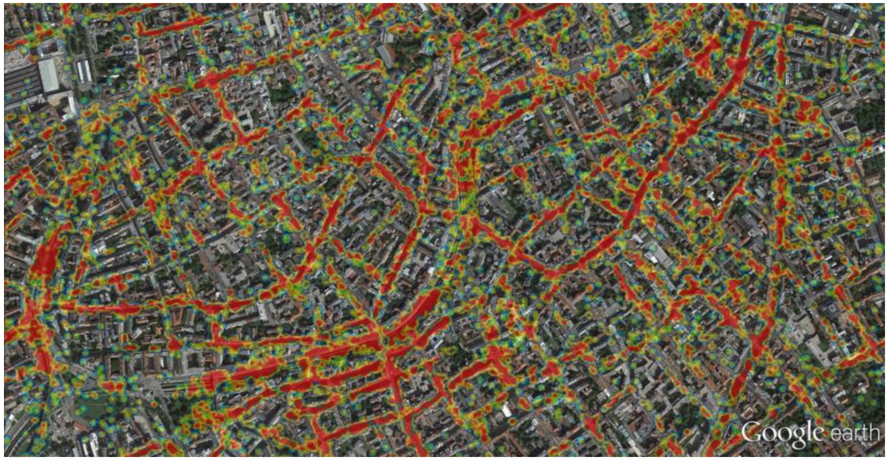
Impact [extended version]

- We used to Wigle war-driving WiFi dataset to identify UPC<n> networks (dataset coverage ?)
- We found 120,380 networks in the city of Vienna alone
- 166,988 networks in Austria including the border region
- We could break into each of them within **3 days at most**

Impact [extended version]



- Density:
  Austria + Border region



- Density:
  City of Vienna

Conclusion

- New implementation speed record
- Professional grade bruteforce speeds are now in the reach of amateurs (e.g. old Bitcoin mining FPGA boards) at a fraction of the cost
- FPGAs are ideally suited for WPA2 cracking
- Real-world networks with weak default passwords can now be broken into within just a few days

Future Work

- Support password lists
- Artix 7 Low-cost cluster
- Evaluation of our implementation on COPACABANA ?

- Thank you for your attention
- More information: read the paper, we recommend the extended version (arXiv:1605.07819v1)
- Contact: Markus Kammerstetter
  <mk _at_ seclab.tuwien.ac.at>

- Questions ?