Reviving smart card analysis

Christopher Tarnovsky chris@flylogic.net
Karsten Nohl nohl@srlabs.de

SECURITY RESEARCH LABS

Executive summary – Modern smart cards should be analyzed

1. Smart card chips provide the trust base for various applications from banking to ID cards to hardware encryption
2. The secured chips not only protect secret keys but also shield software and protocols from independent analysis. Vulnerability analysis by manufacturers and contracted labs has overlooked bugs numerous times, so independent analysis is needed for software protected by smart cards
3. This talk describes a method for extracting smart card software for analysis through semi-automated reverse-engineering of circuits and optical memory read-out
4. The potential for code read-out affects most modern smart cards. Mitigating it requires stronger memory encryption combined with smarter key storage

Agenda

- **Smart card basics**
- Reverse-engineering memory encryption
- Mitigation measures

A few smart cards chips cover numerous security domains

Smart card applications

Payment cards, electronic IO cards, access badges

Trusted Platform Modules (TPM)

Device and accessory identification

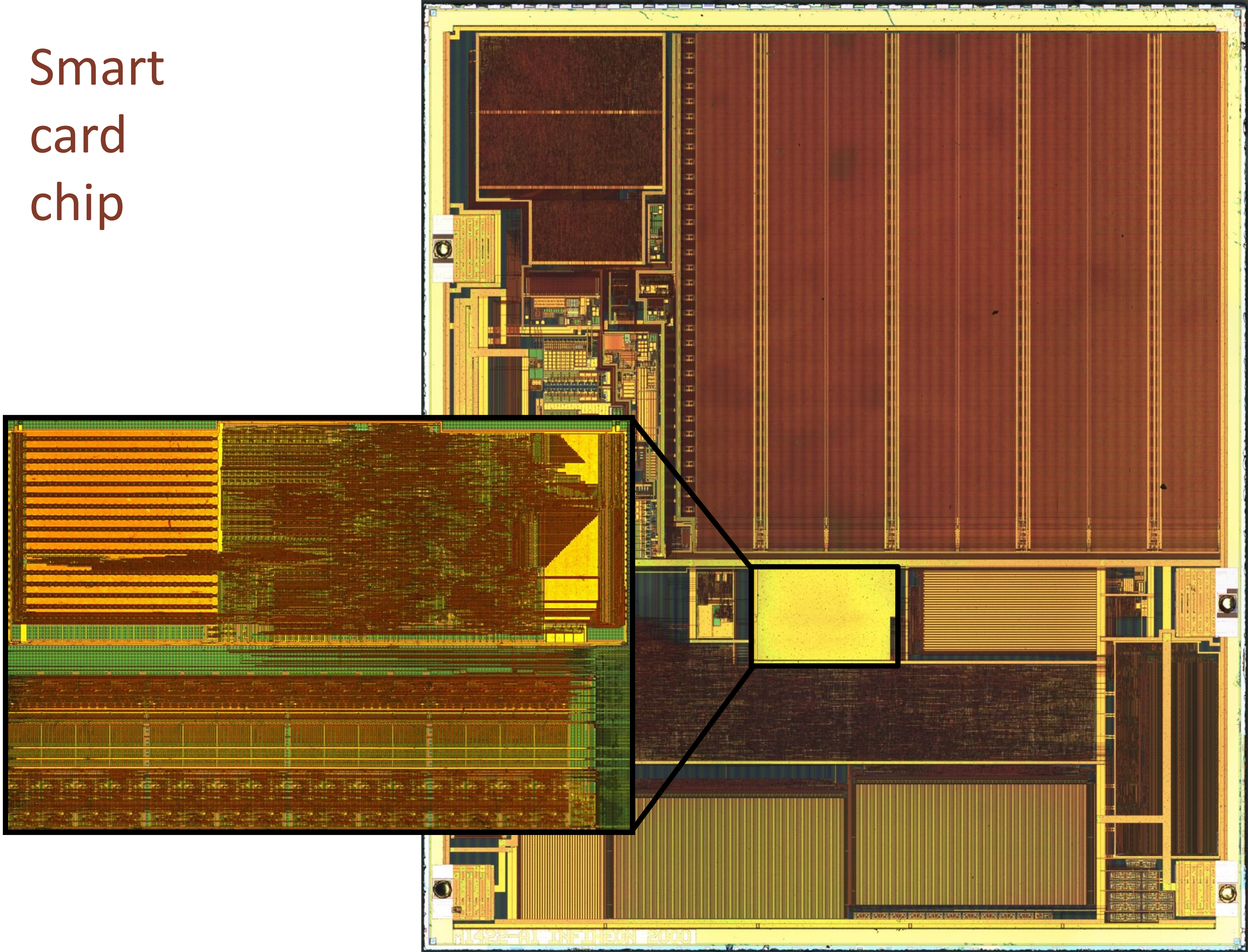
SIM cards, NFC secure elements

Content protection

Bulk risk

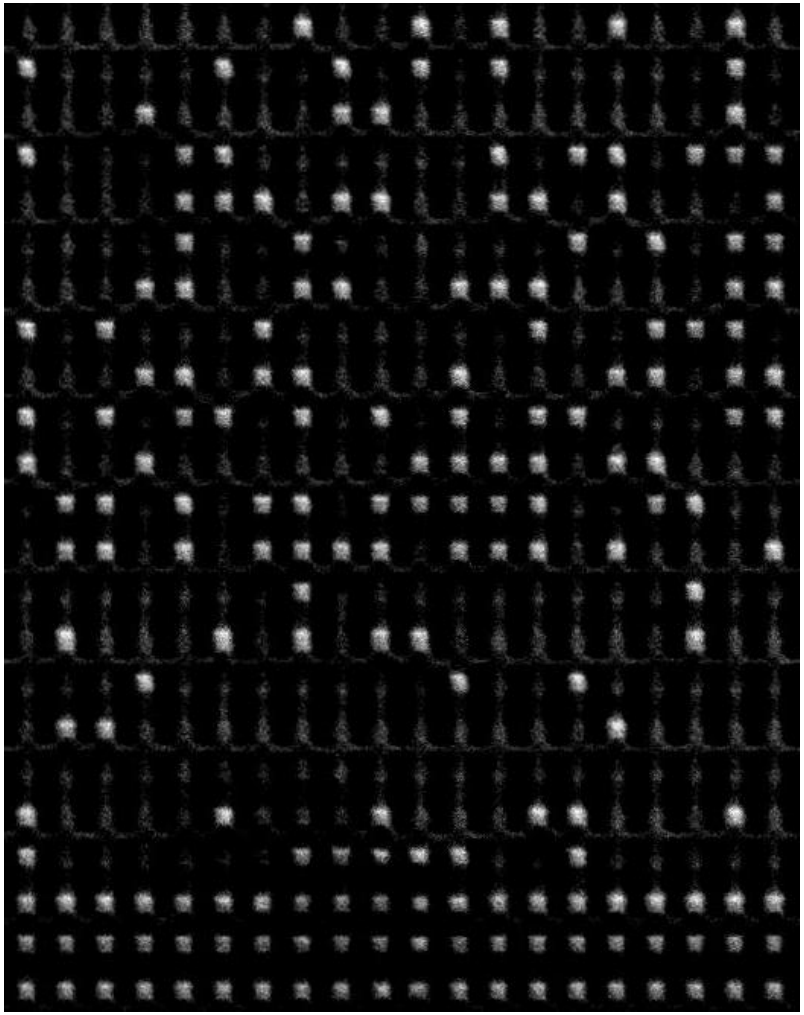
- Only a handful of manufacturers provide smart cards
- Therefore, every card compromise affects numerous applications

Smart card chip



Unencrypted ROM memory can be read optically

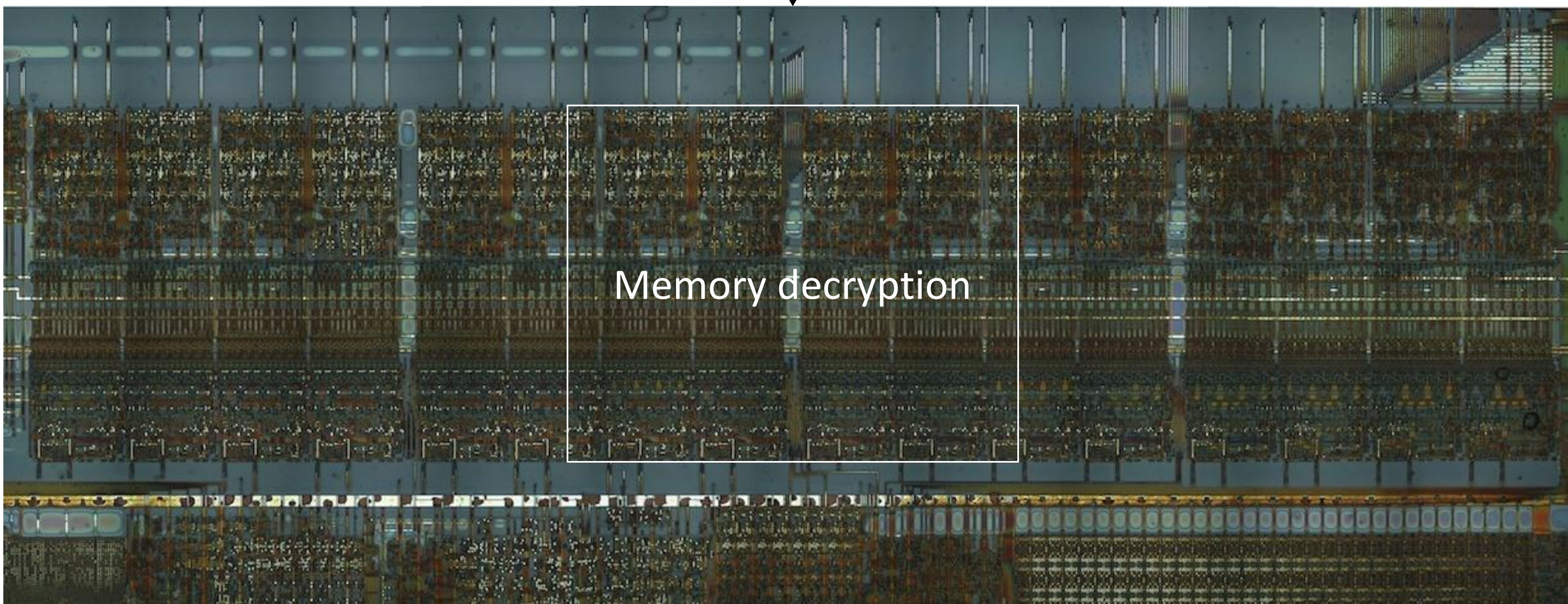
Metal-masked ROM



Ion-implanted ROM (after staining)

Smart cards protect code and data with memory encryption

Memories



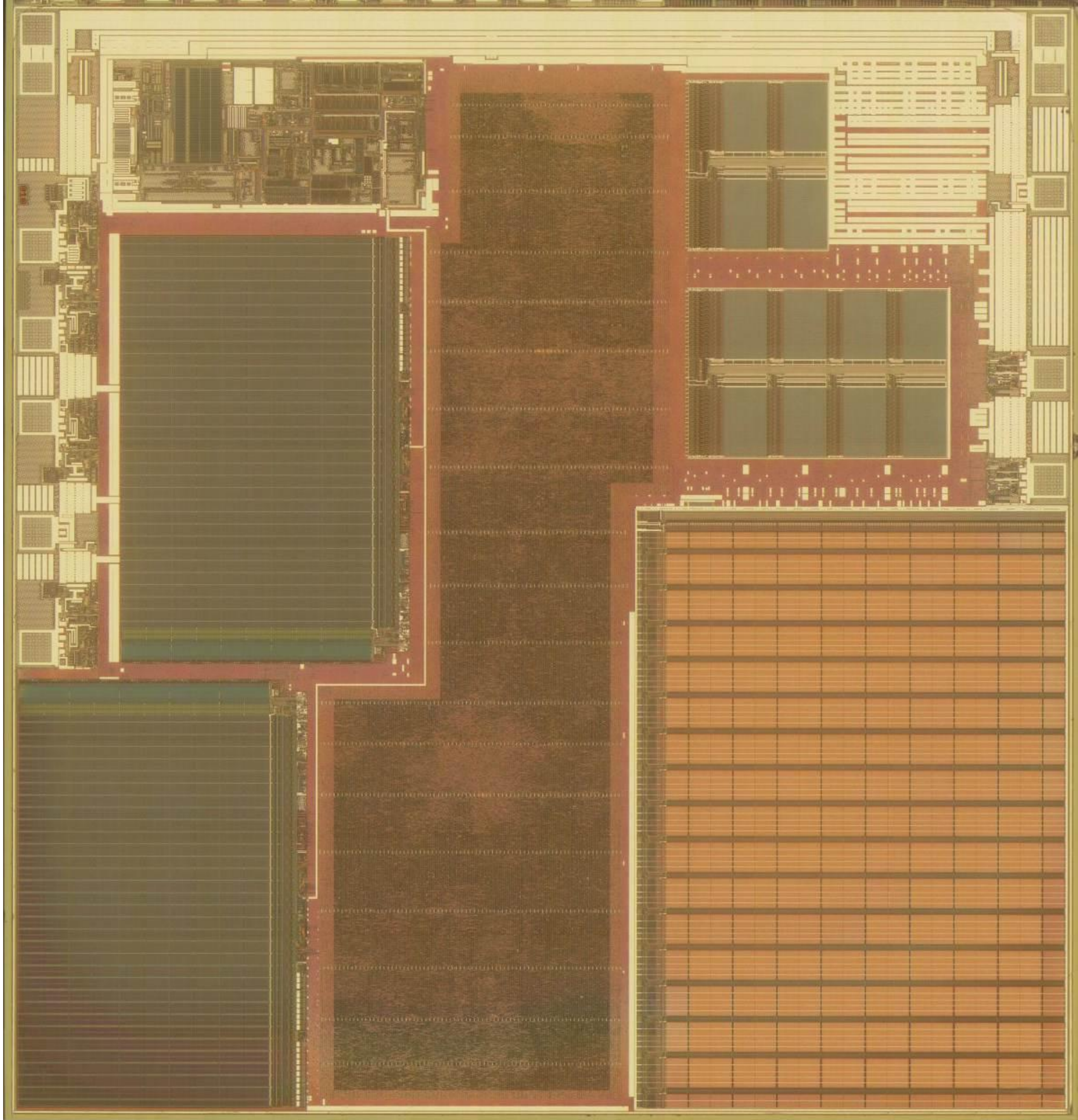
CPU

Newer smart cards hide the decryption in the CPU

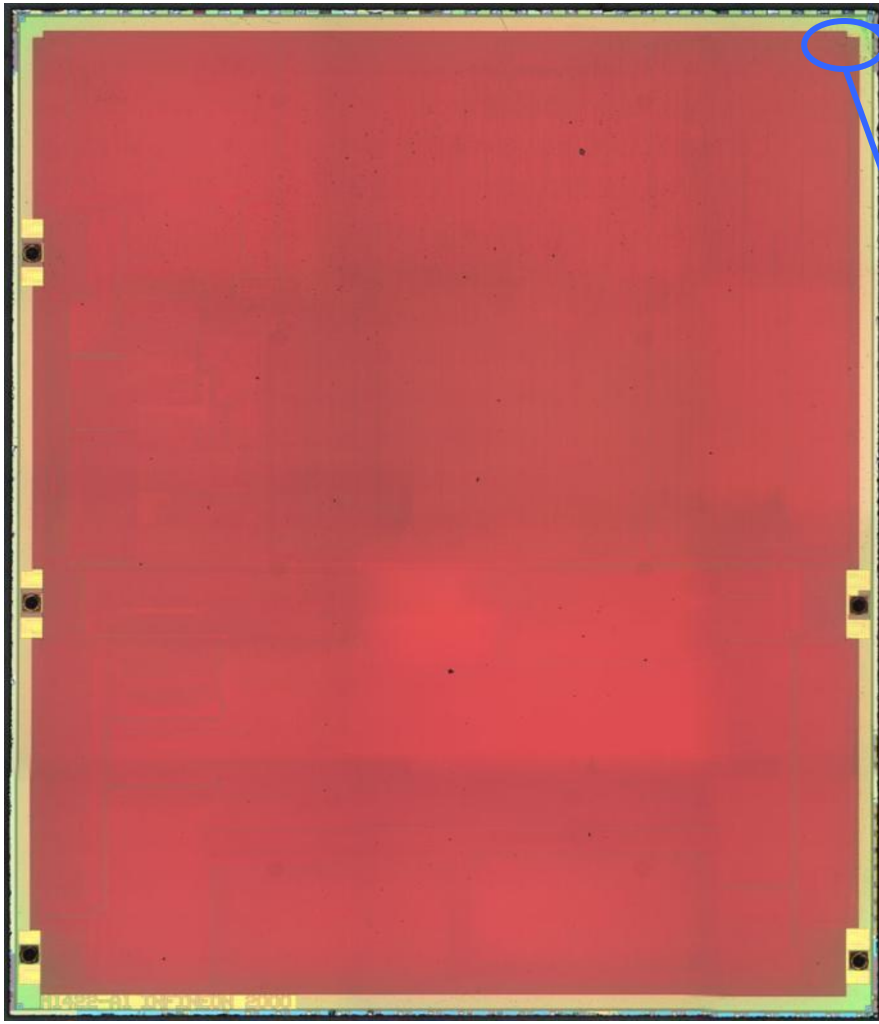
Recent design iterations moved cryptography deeper into chips

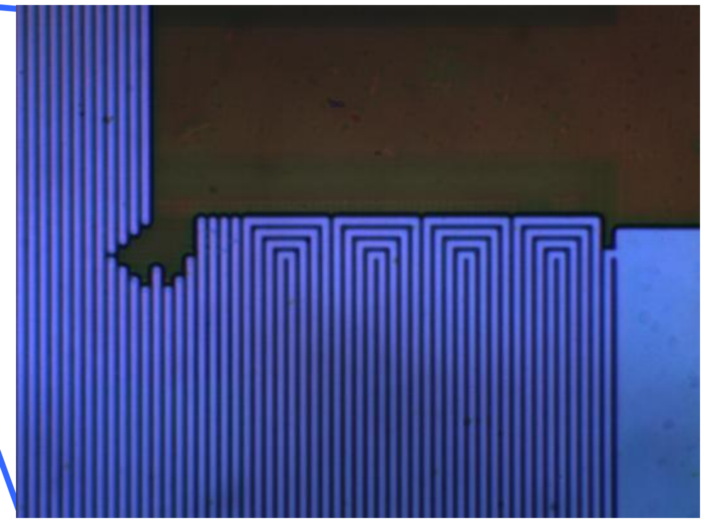
	1	2	3
Attack	Optically read code and data from memories	Probe wires after MED that carry unencrypted data	Reverse-engineer (part of) CPU to find unencrypted wires
Mitigation	Encrypt memories on the fly in MED	Move encryption into CPU	Fully encrypted CPU (possible?)

Smart card with memory encryption in CPU core



Protection meshes create additional complexity for invasive attacks



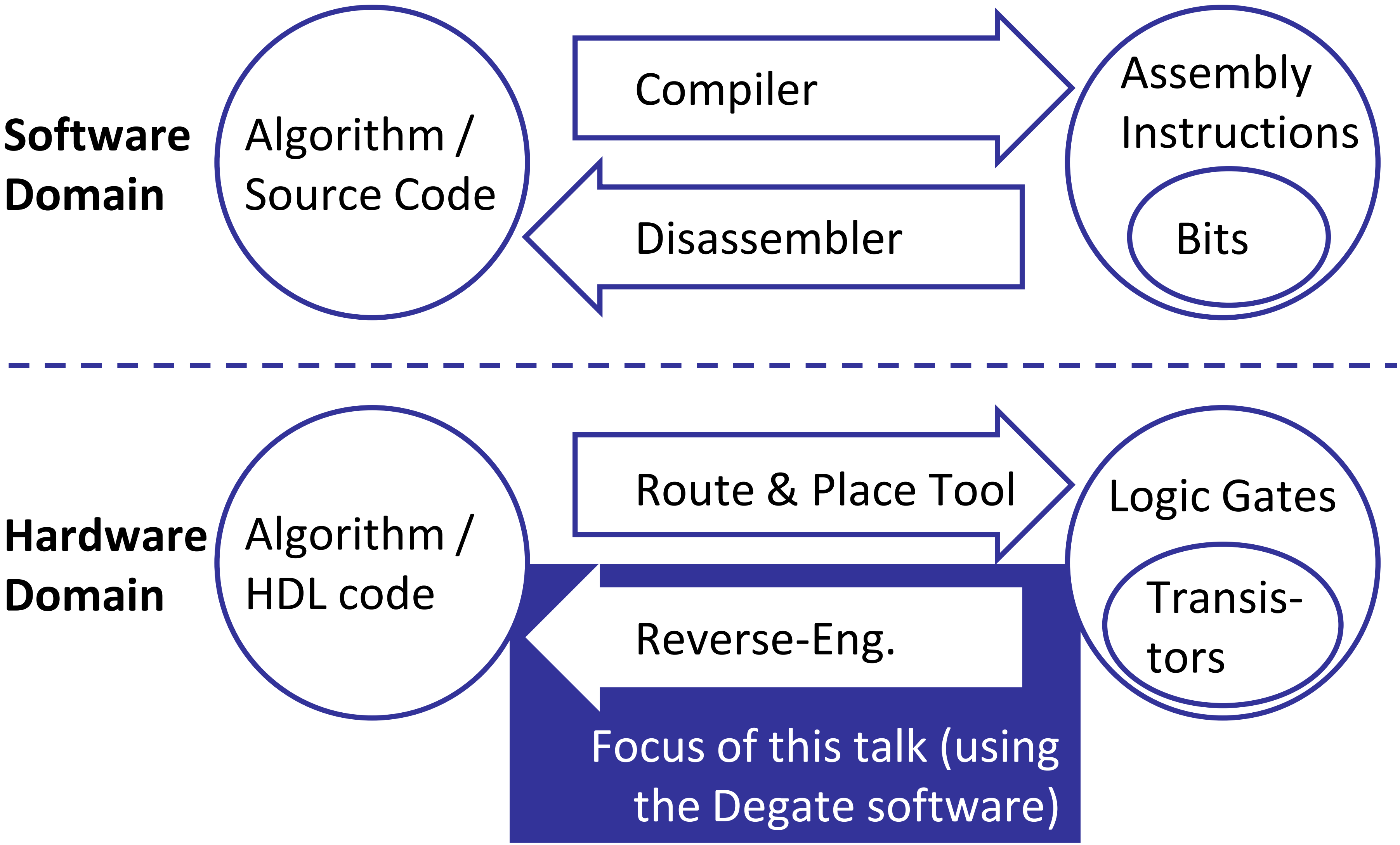


- Meshes increase effort of invasive data extraction
- This talk introduces method for extracting ROM code not hindered by meshes

Agenda

- Smart card basics
- **Reverse-engineering memory encryption**
- Mitigation measures

Researchers need better tools for finding algorithms in hardware



Algorithms can be extracted from chips in a 3-step process

Silicon disassembling process

Image chips
- Polishing
- Microscoping
- Stitching

Recognize structures
- Pattern recog.
- Wire tracing

Interpret structures
- Gate simula-tion

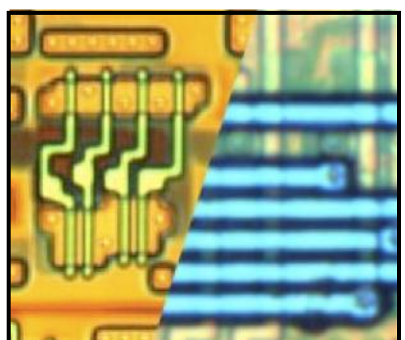
Algo-rithm

Silicon Chip

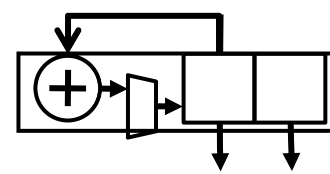
Chip images

Netlist

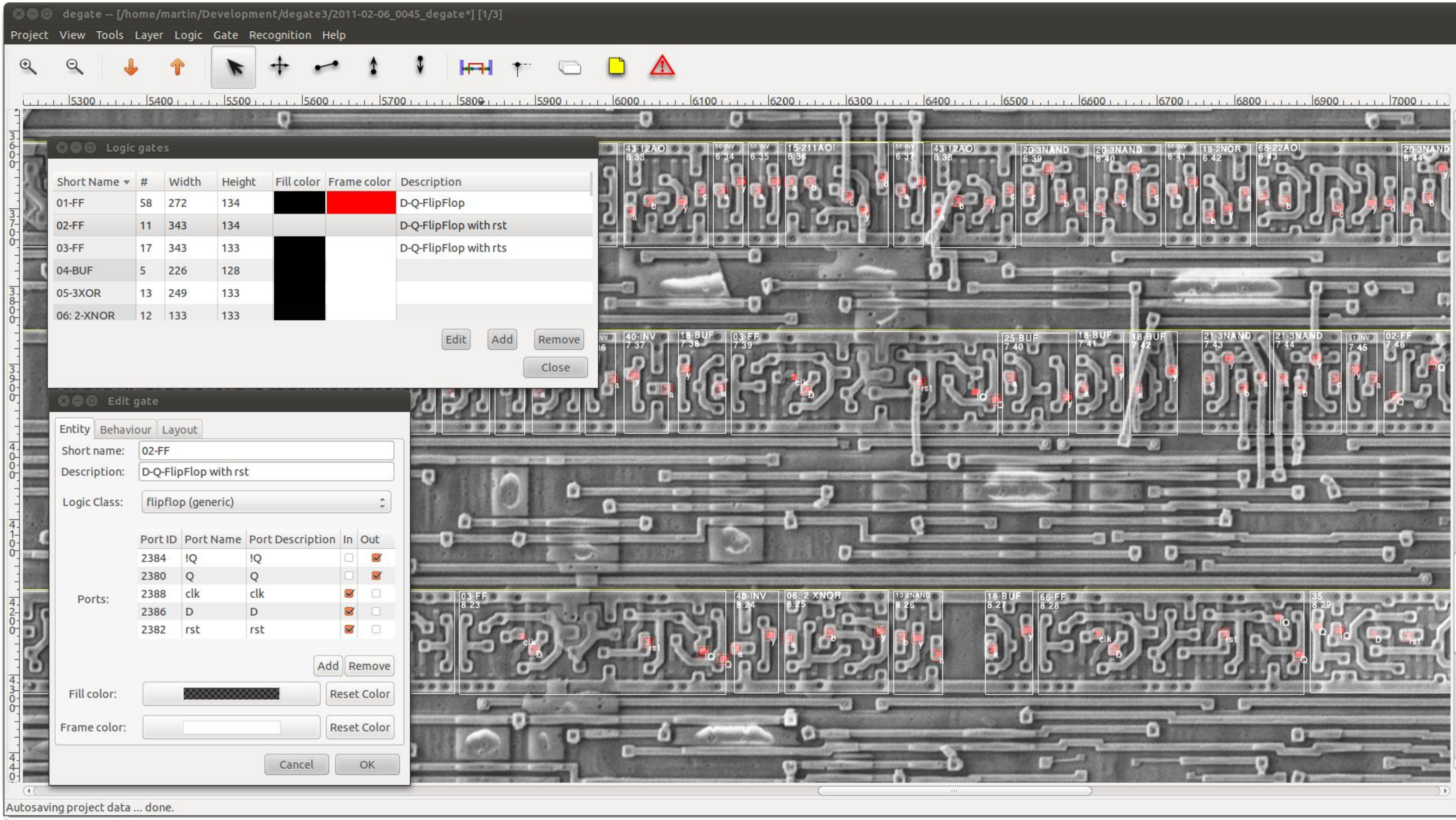
Annotated netlist



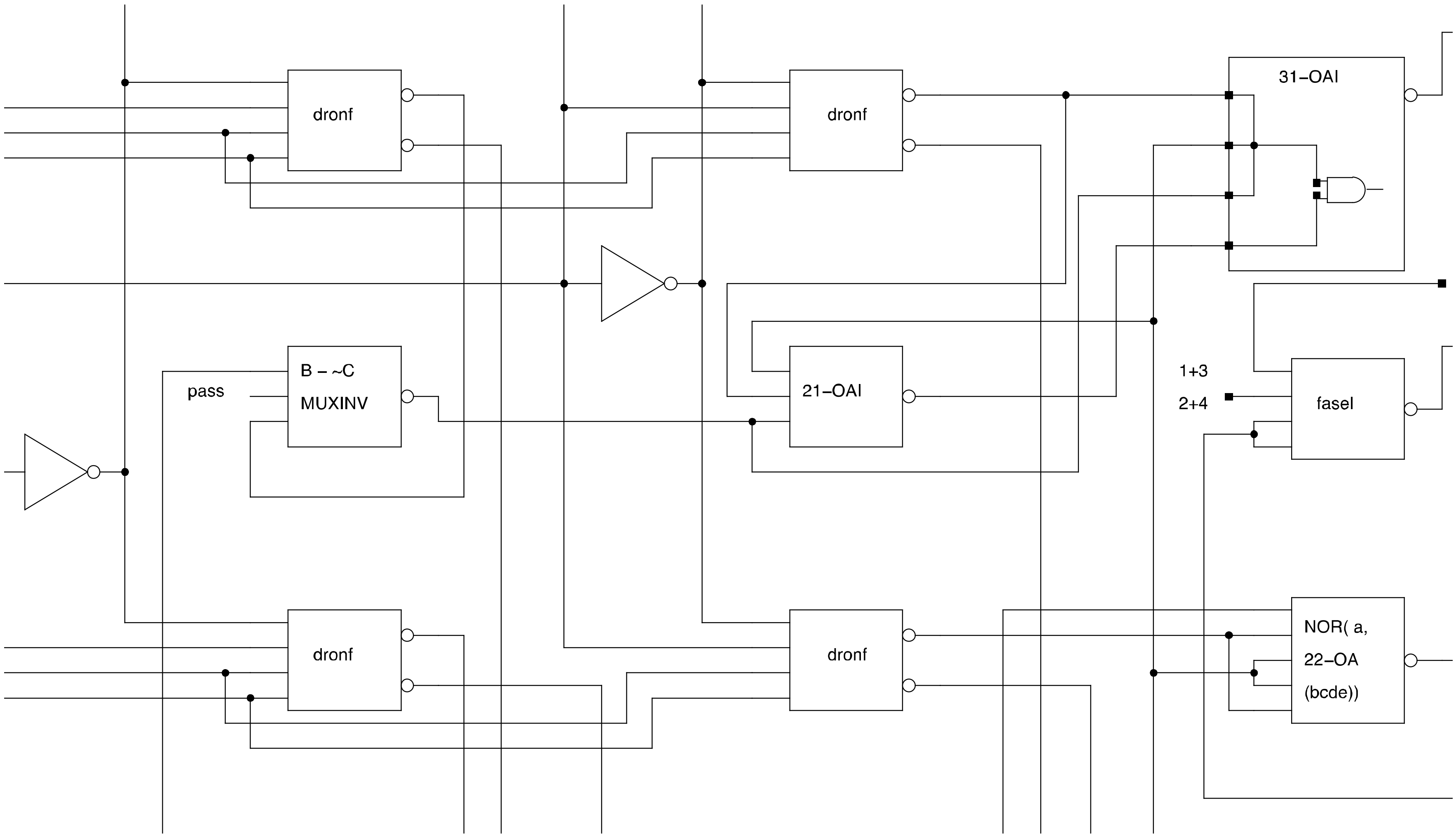
```
<?xml version="1.0"
<gate-library>
 <gate description=
  <ports>
   <port id="0" nam
   <port id="1" nam
```



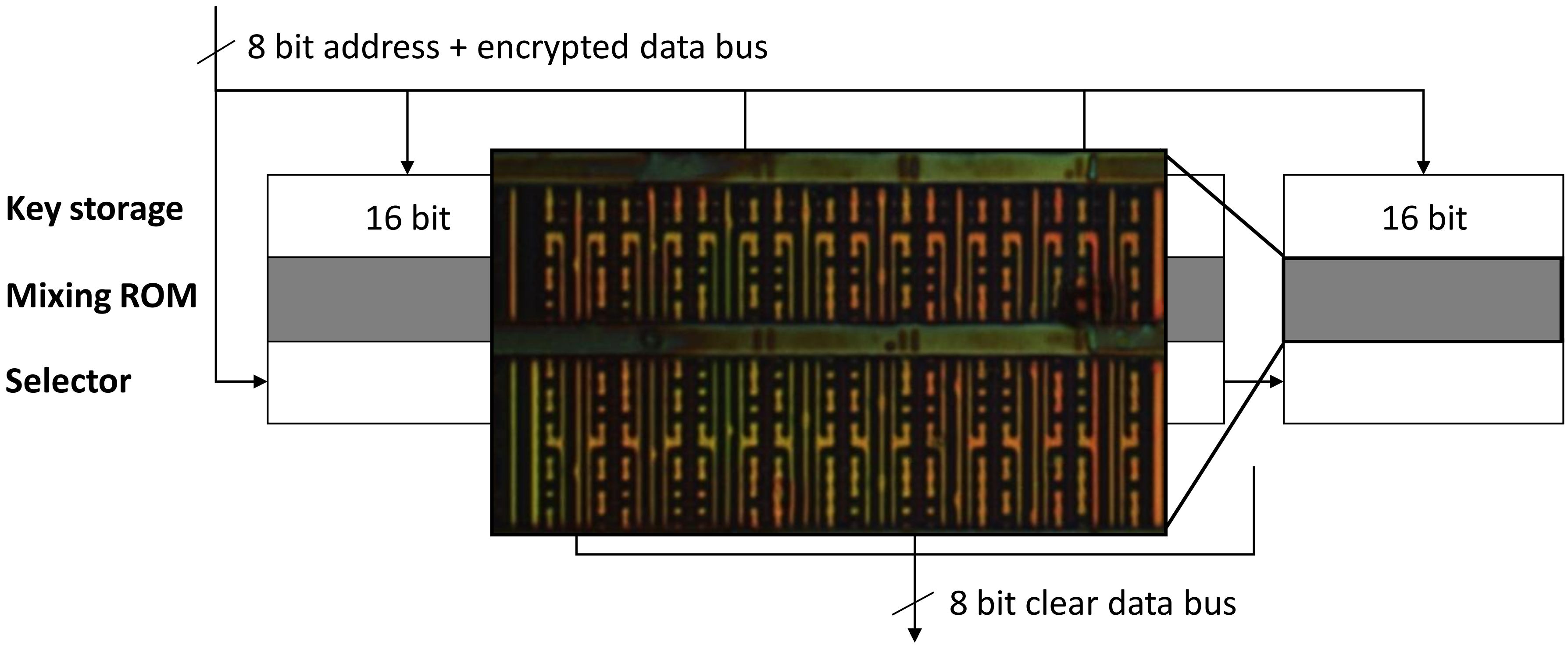
Demo: Reverse-engineering with Degate



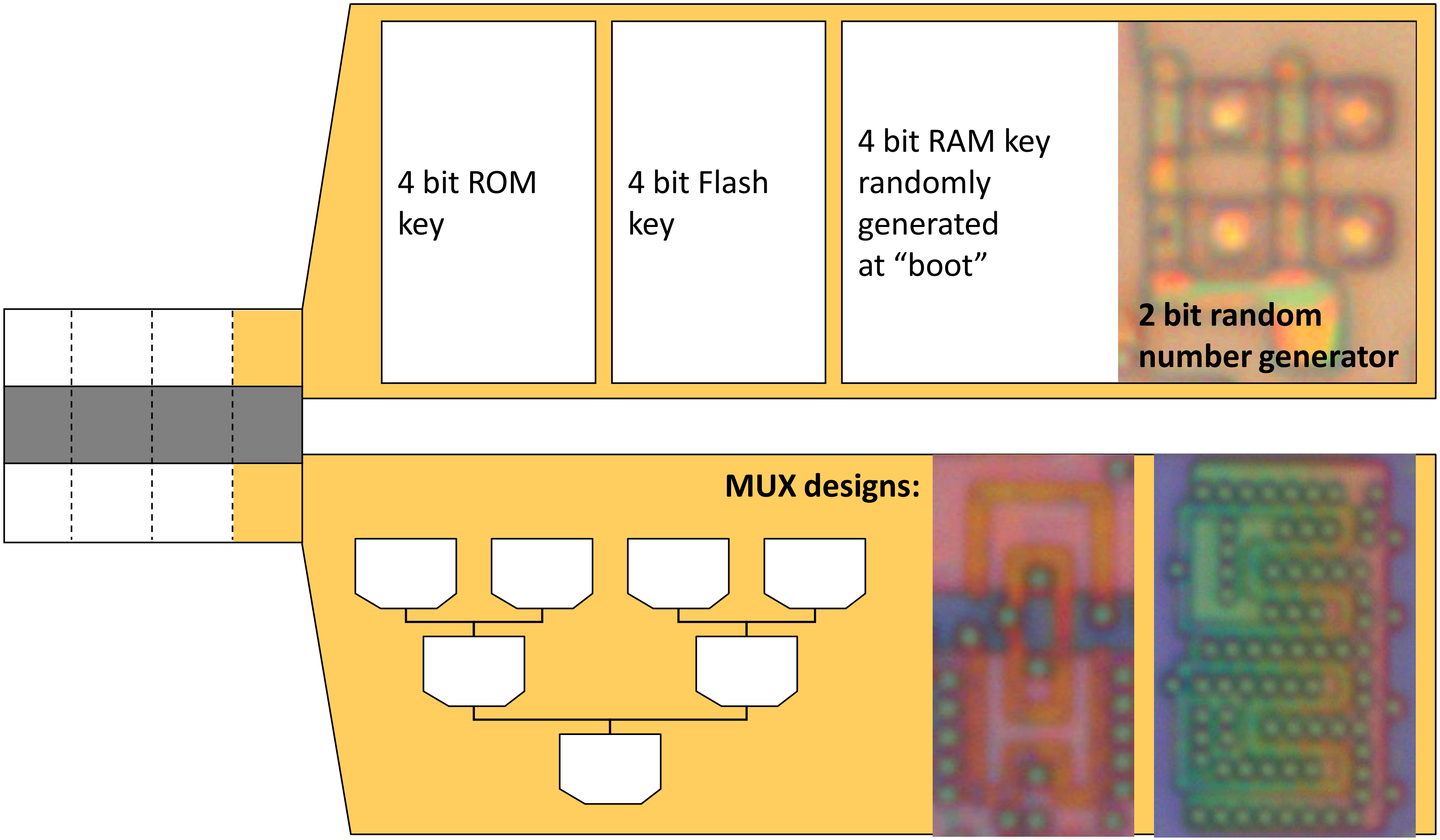
Degate outputs synthesizable code that can be visualized with standard tools



Memory is ciphered with 64 bit key



Memory encryption includes non-deterministic and obfuscated cells



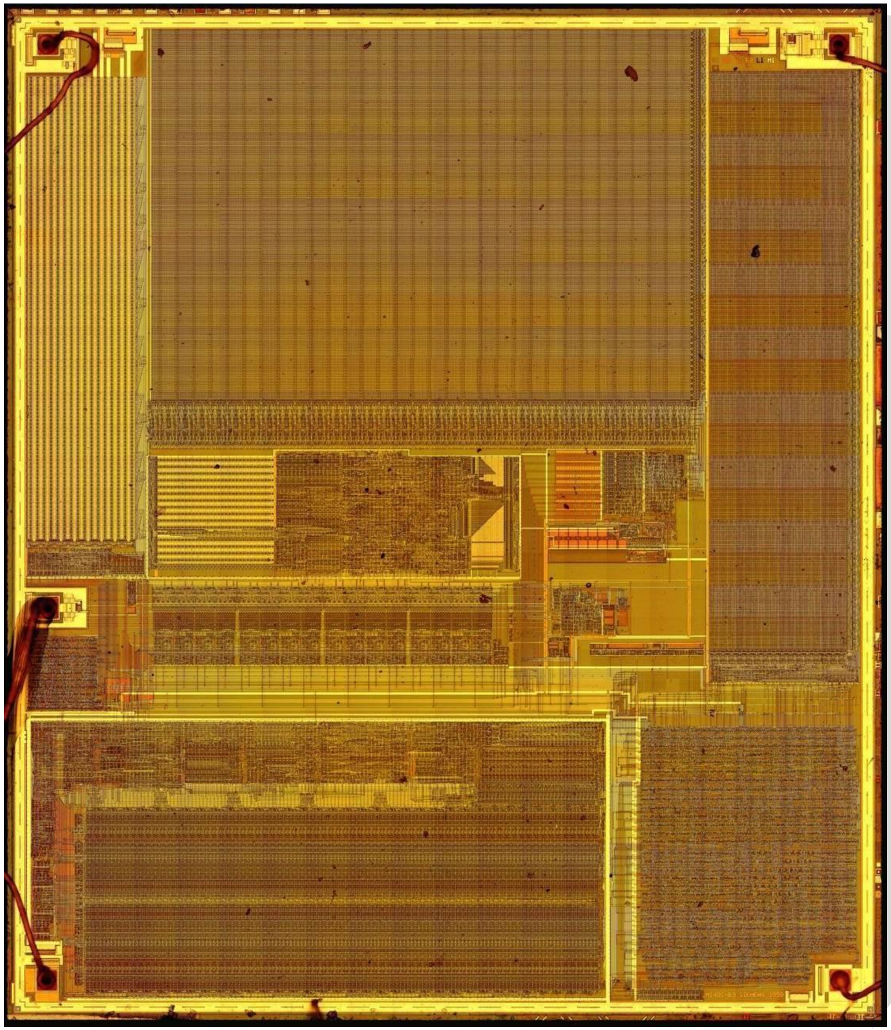
Agenda

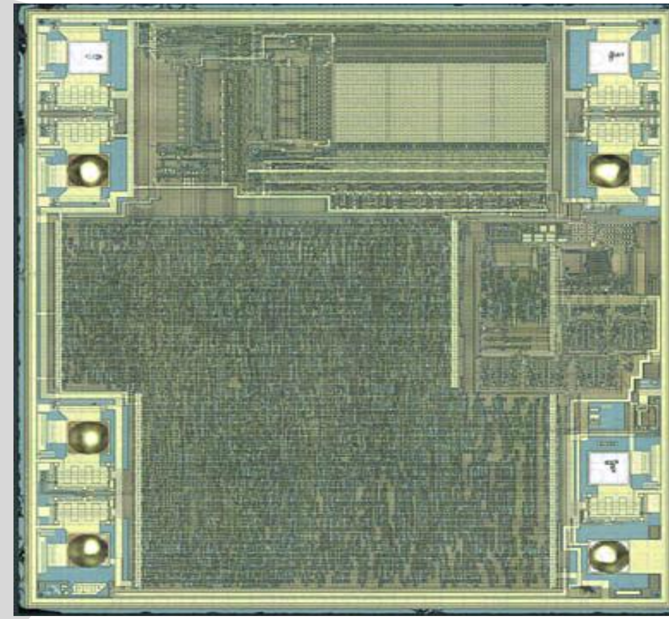
- Smart card basics
- Reverse-engineering memory encryption
- **Mitigation measures**

Hardening smart card systems requires actions from manufacturers and system designers

Mitigation step	Responsible
A Increase complexity of reverse-engineering	Smart card manufacturer
B Harden cipher against cryptanalysis	
C Keep smart card attack value low	System designer

A Best defense against reverse-engineering – Increase complexity, decrease feature size





- Use more random logic, separated in fewer blocks
- Make it harder to image chip (feature size <180 nm)

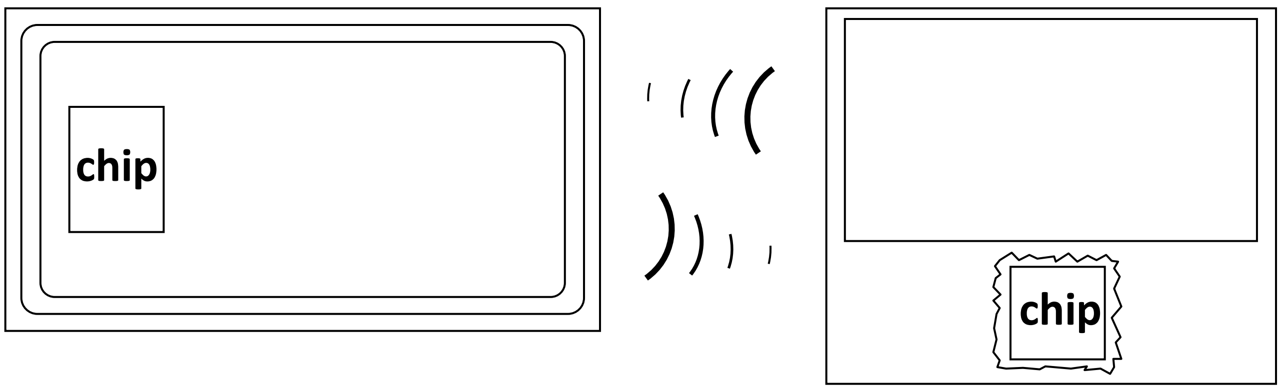
B ROM code could be protected through better cryptography

Outlined attack	Possible mitigation
1 **Optical ROM read out (including decryption key!)**	Store ROM key in hard-to-read memory (Flash, PUF) **Effort: Easy** to implement on modern chips that have Flash
	+
2 **Reverse-enginee-ring of proprietary decryption function**	Use well-designed crypto-graphy to prevent key cracking **Effort: High,** research into lightweight crypto needed!

Technology risks vary widely with use case

Example: Nationwide micro-payment scheme

Payment card | Payment terminal

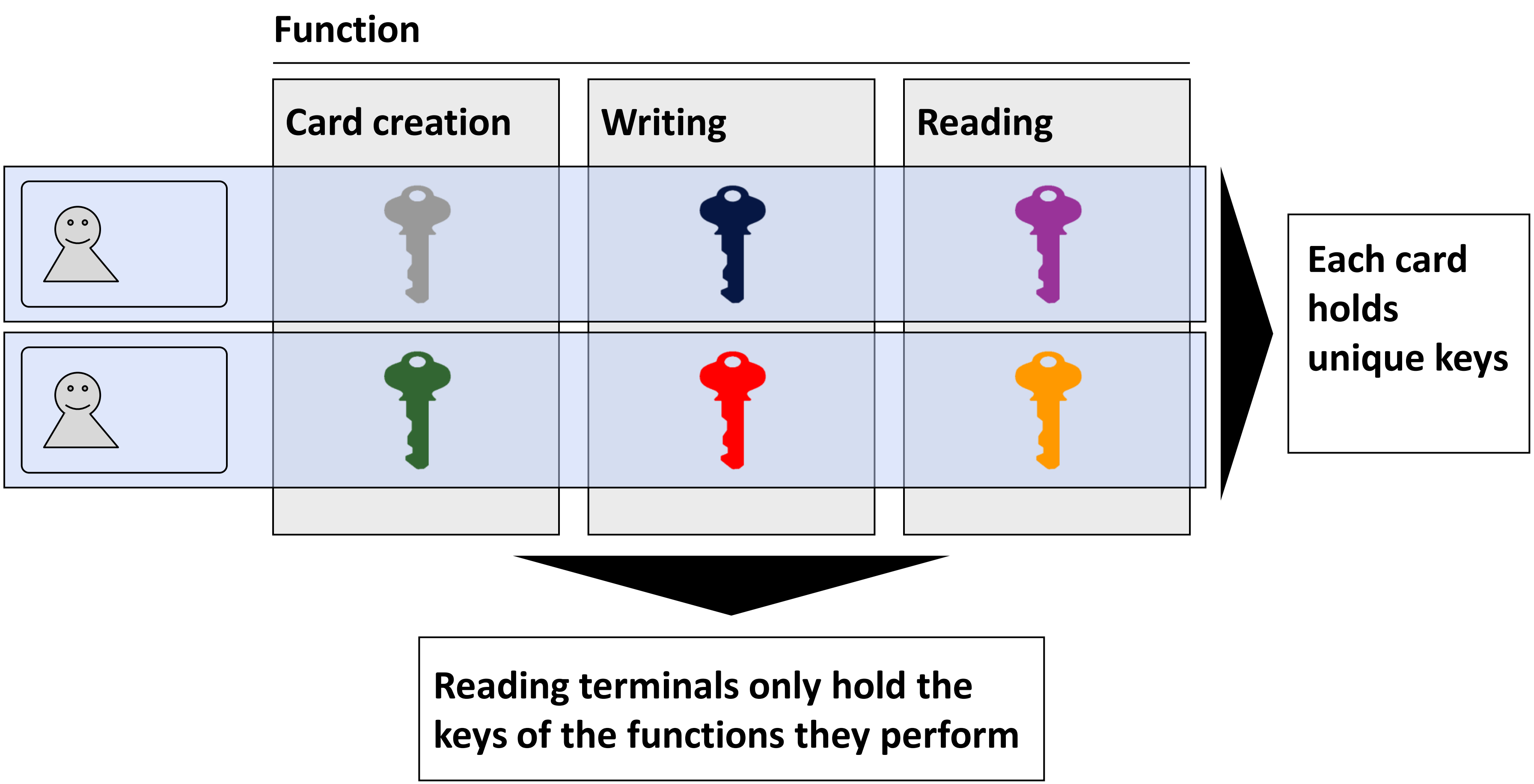


Extracting secret keys allows cloning **one card**

Extracting secret keys allows cloning **all cards**

Same chip, same protection, but different security level

C Key distribution should follow need-to-know principle to limit attack surface



Smart card can provide large but not infinite level of protection

All smart cards can be reverse-engineered or broken with intrusive attacks at some cost

Software stored in ROM should be extractable for analysis from most modern smart cards

Smart cards are the most secure hardware, but system designers must limit attack incentives and not expect miracles from chips

Degate software, degate.org
Silicon gate examples, siliconzoo.org

Questions?
Karsten Nohl, nohl@srlabs.de
Chris Tarnovsky, chris@flylogic.net