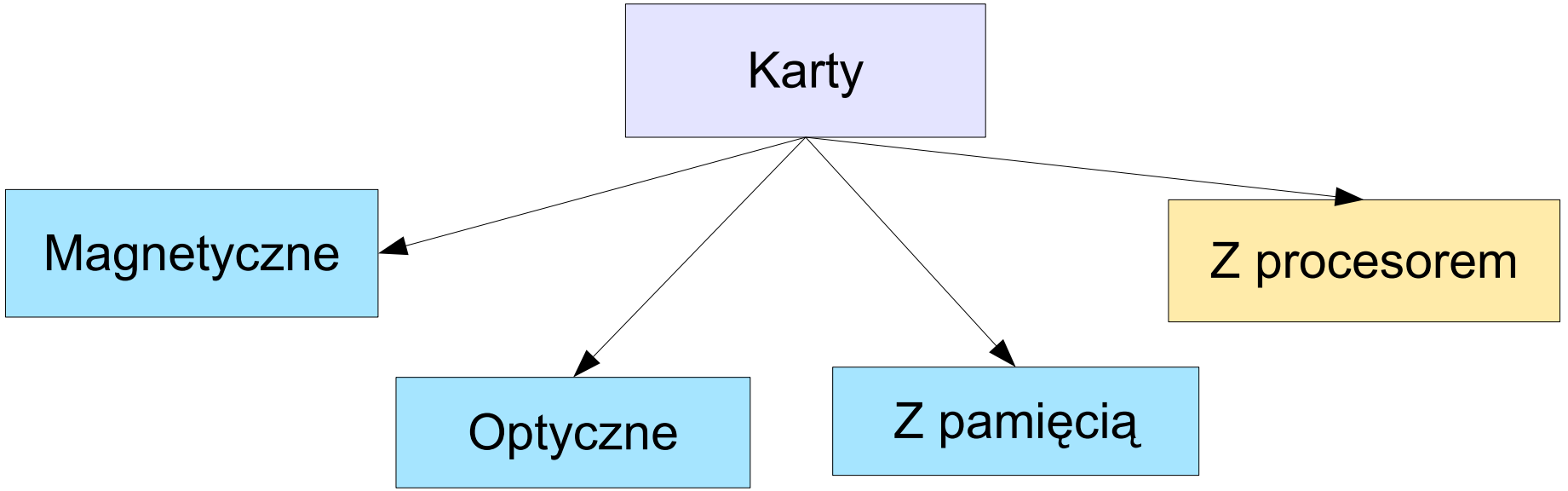
Bezpieczeństwo kart inteligentnych

- Co to jest karta inteligentna?
- Krótka historia, technologie i standardy.
- Modelowanie bezpieczeństwa kart.
- Wybrane ataki na dane.

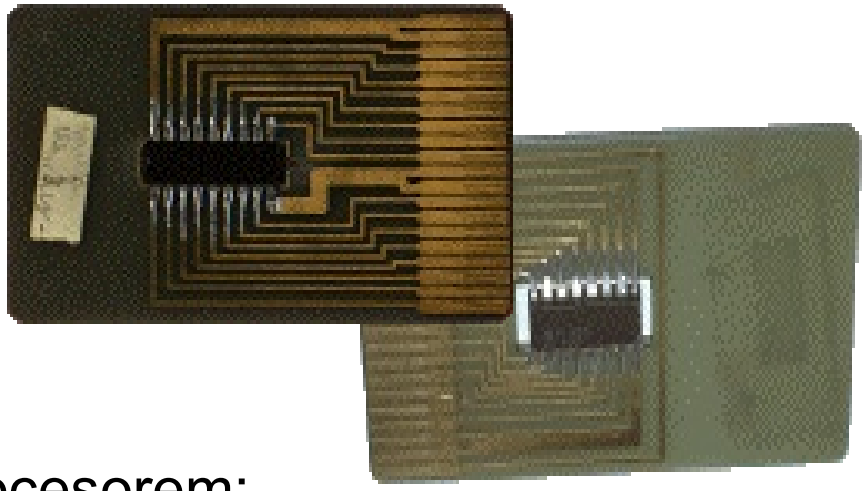
Karta inteligentna (*smart card*)

Karta wielkości kieszonkowej z wbudowanym układem scalonym.

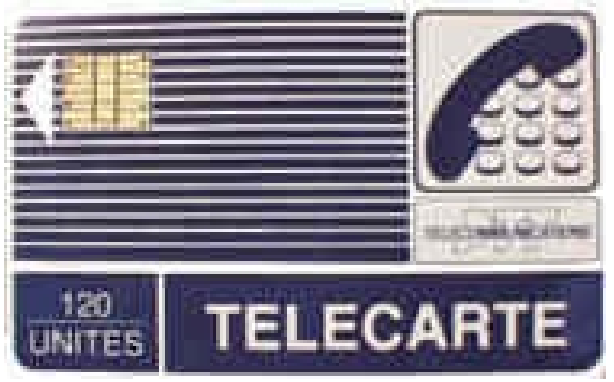


Główna motywacja – brak możliwości zabezpieczenia kart magnetycznych.

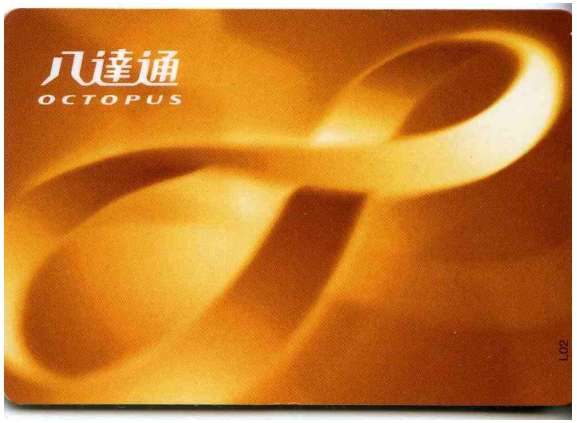
Krótka historia kart



- 1974 – pierwszy patent dotyczący kart z pamięcią (Roland Moreno, Francja)
- 1977 – pierwszy patent na kartę z mikroprocesorem; banki francuskie tworzą specyfikację karty płatniczej, rok później powstaje pierwszy prototyp;



- 1983 – pierwsze masowe użycie – francuskie płatne telefony, *telecarté*
- 1986 – 14000 kart wydanych klientom *Bank of Virginia* i *Maryland National Bank*; trochę później 50000 kart wydanych klientom dwóch innych banków



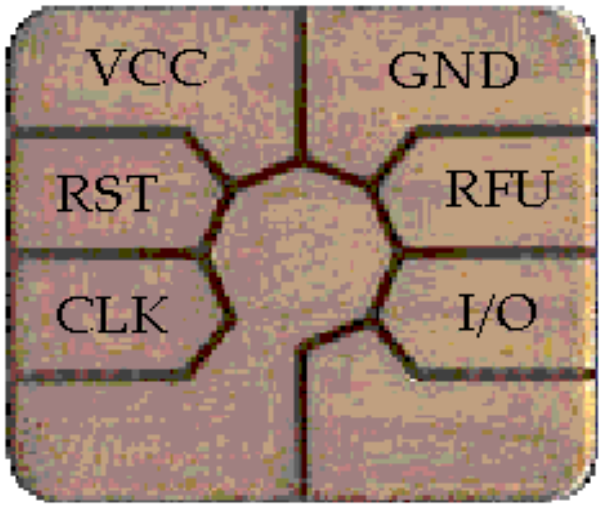
- 1992 – *Carte Bleue* we Francji – mikroprocesory we wszystkich kartach debetowych; pierwszy (narodowy, przedpłaty) system elektronicznych pieniędzy (*Danmont*, Dania)
- 1993 – pierwszy test karty wielofunkcyjnej (*Smart Bank Card* + *telecarté*)
- 1994 – *Europay*, *MasterCard* i *Visa* publikują wspólny standard dla kart bankowych (*EMV*); w Niemczech rozpoczyna się wydawanie 80 milionów inteligentnych kart zdrowia
- 1995 – karty w telefonach komórkowych
- 1996 – ponad 1.5 milionów kart – elektronicznych pieniędzy – na igrzyskach w Atlancie
- 1997 – pierwsza karta bezkontaktowa – *Octopus* w Hong Kong'u
- 1999 – pierwszy system praw jazdy oparty o karty – Gujarat w Indiach

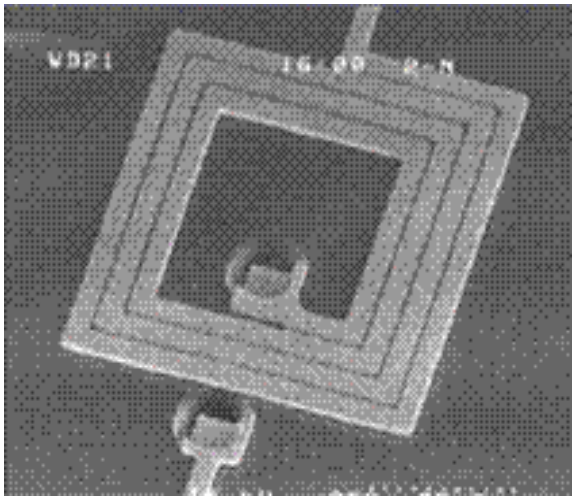
Zastosowania

- Kilkaset milionów telefonów z kartą w środku.
- Karty telefoniczne z chipem.
- Komunikacja publiczna.
- Karty w dekoderach telewizyjnych.
- Karty bankomatowe, debetowe itp..
- “Stored Value Card” - jedna z odmian elektronicznych pieniędzy.
- Karty dostępu do budynków i usług.
- Karty jako element uwierzyteniania.
- Książeczki zdrowia, prawa jazdy i inne dokumenty.

Karty z kontaktami

- standardy: ISO 7810 – 7816, EMV 4.1
- zasilanie: 5V
- zegar: 3.5MHz

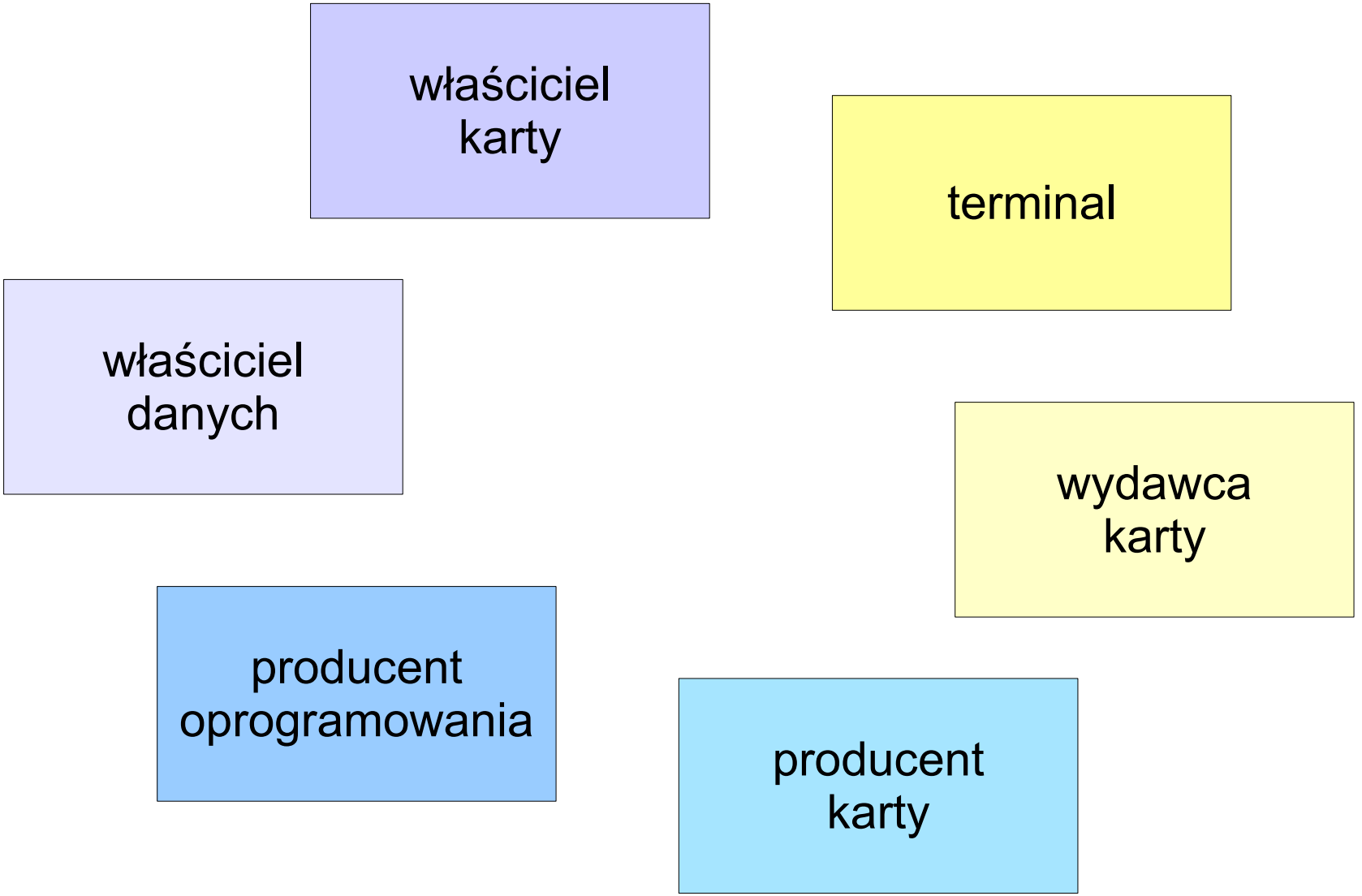




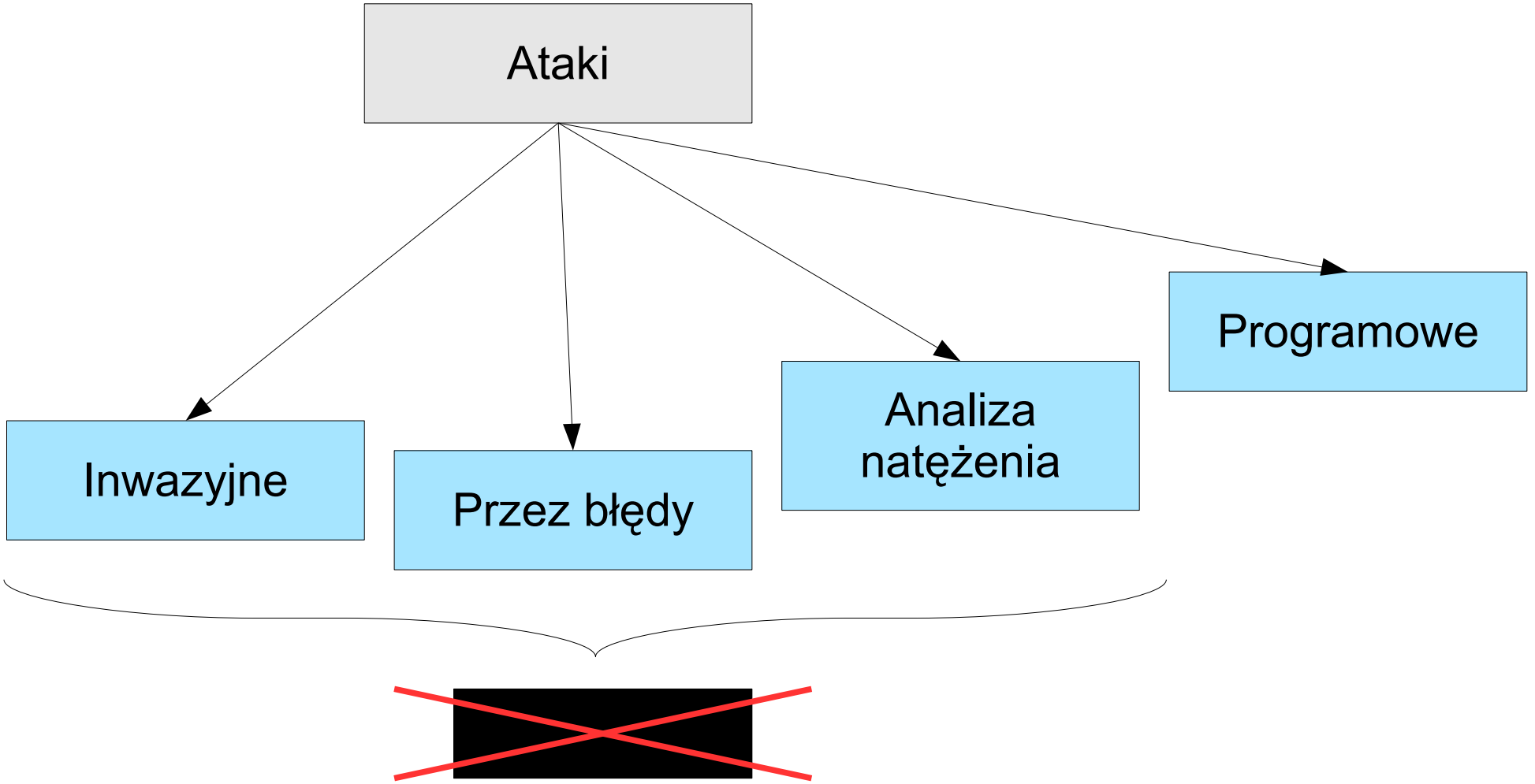
Karty bezkontaktowe

- standardy: ISO 14443,15693
- pole magnetyczne: 1.5 – 7.5 A/m
- komunikacja do 10 cm, ewentualnie 50 cm
- transfer: 106 – 848kb/s
- zegar: 13MHz

Model bezpieczeństwa

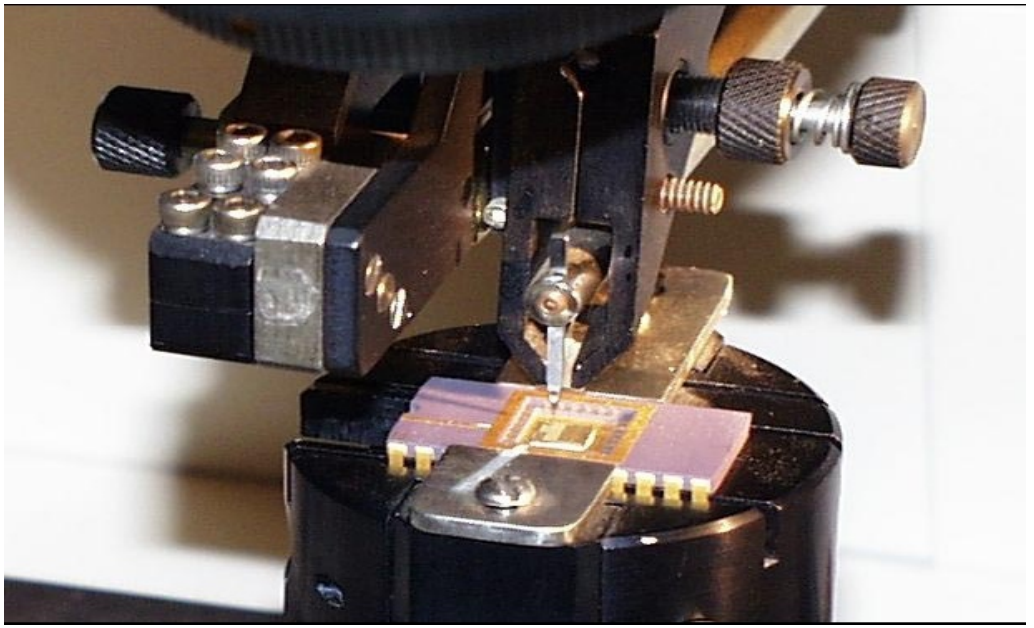


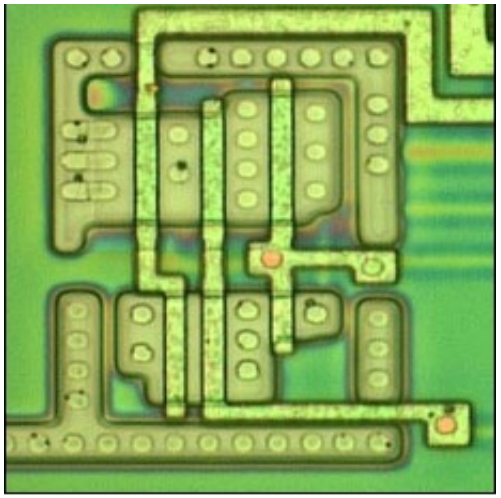
Rodzaje ataków przeciw danym

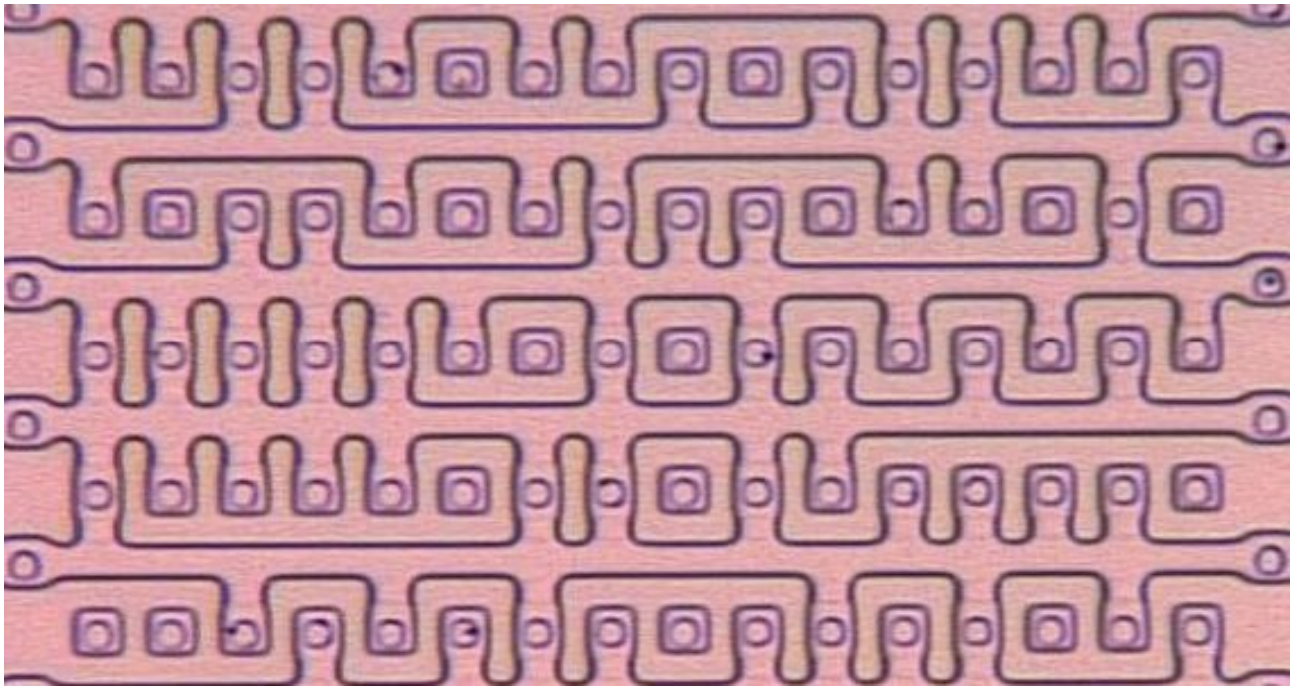


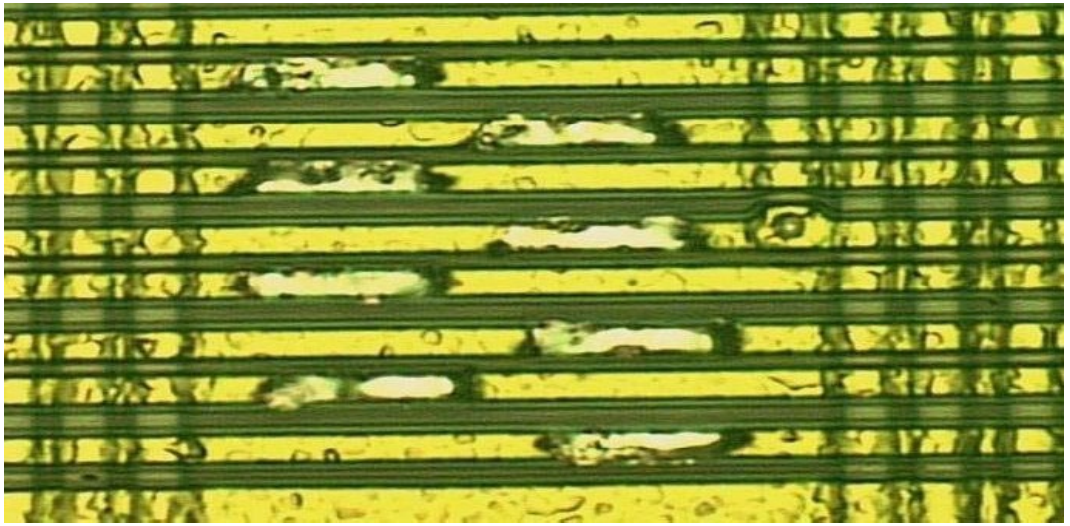
Metody inwazyjne

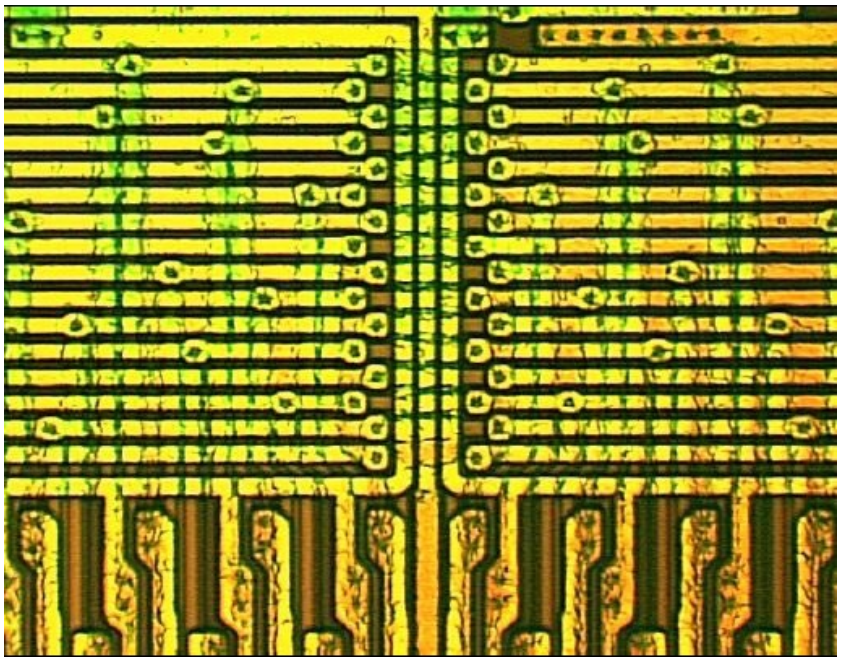
- Inaczej: Mikropróbkowanie
- Niszczymy kartę
- Potrzebne laboratorium – duży koszt
- Nie trzeba znać architektury
- Nie ma odpornych kart
- Często najpierw taki atak – jeden raz, a później inne – tańsze metody



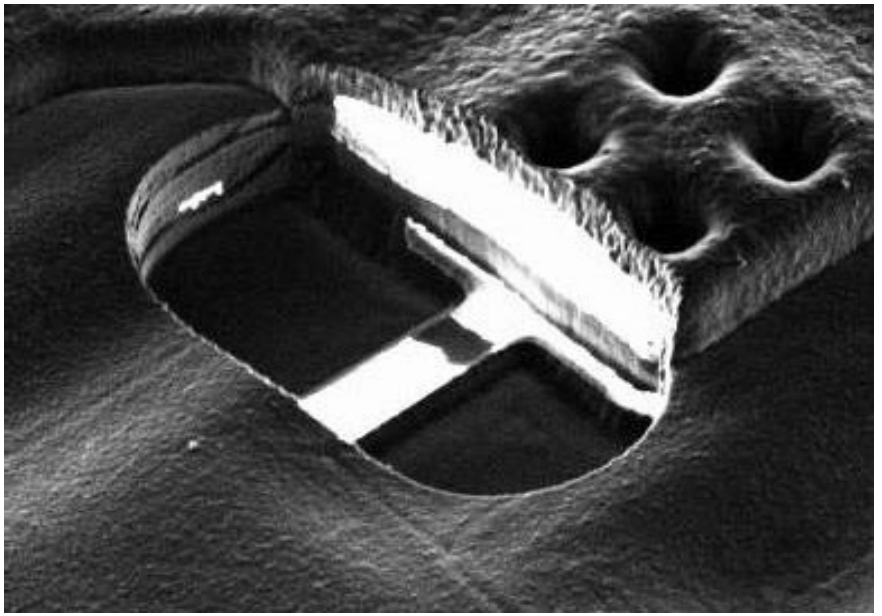




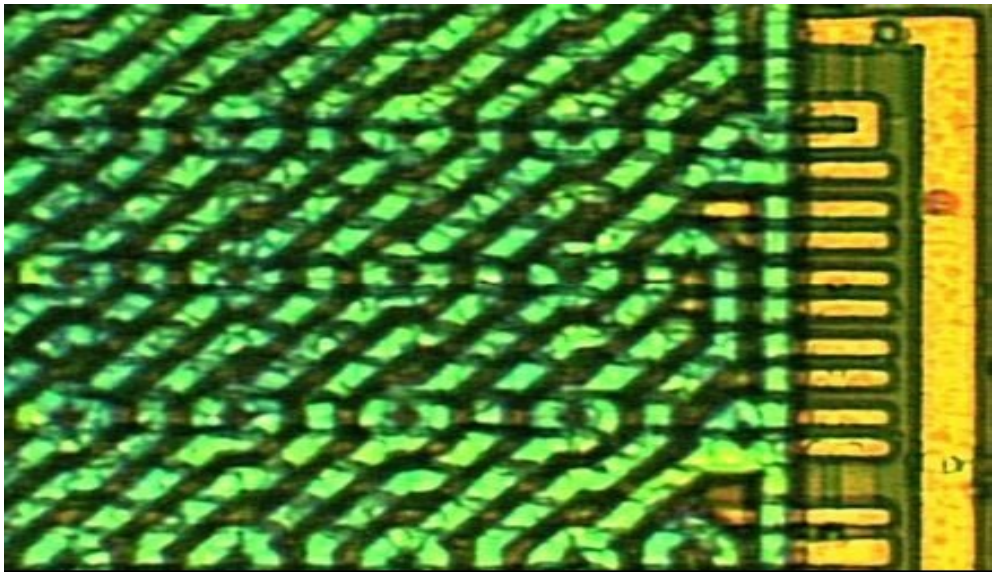


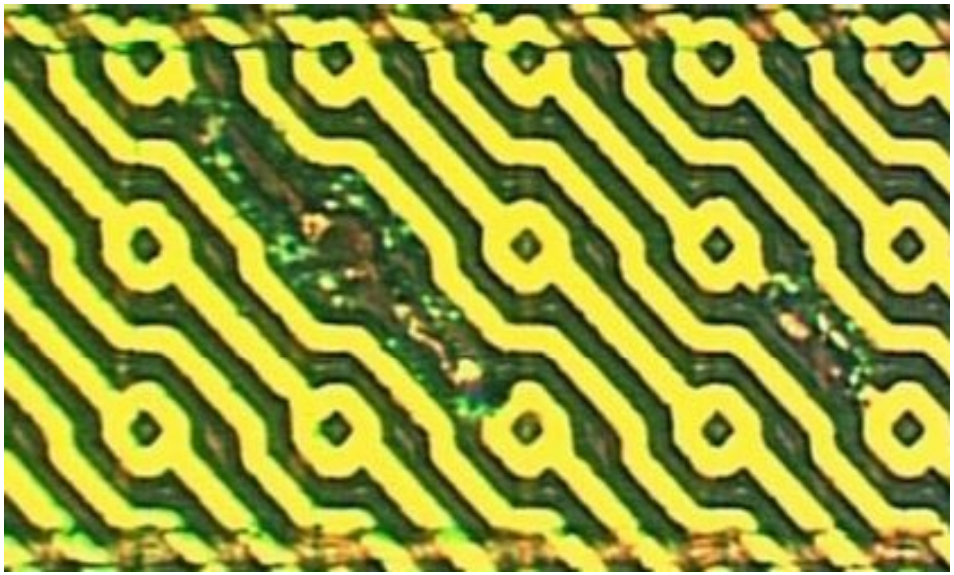


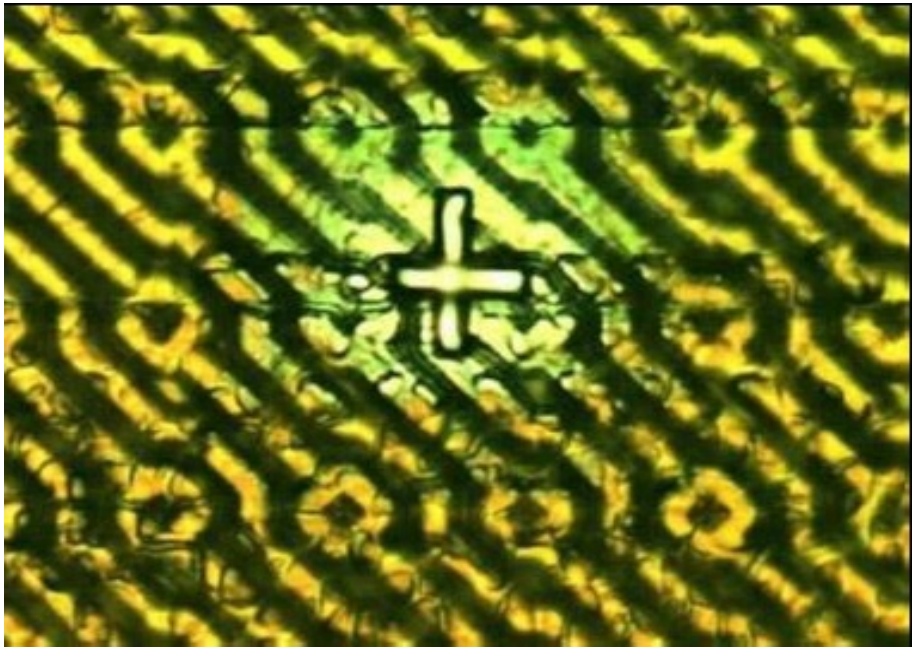
Przeciwdziałanie



- Niszczenie układów testowych.
- Wierzchnia warstwa sensorów.
- Dobry czujnik niskiej częstotliwości.
- Zabezpieczenia fizyczne.
- Nie używanie standardowych układów.







Metody nieinwazyjne

Analogowe efekty, które można wykorzystać:

- Każdy tranzystor ma pewną pojemność i oporność, które razem z temperaturą, napięciem itp. determinują szybkość propagacji sygnału.
- Rejestr próbkuje swoje wejście w krótkim przedziale ustalonym względem taktu zegara; porównuje napięcie z napięciem zasilającym.
- Bramki akceptują nowy stan dopiero gdy wyjścia logiki ustabilizują się na poprzednim stanie.
- W bramce CMOS, przy każdej zmianie wartości, przez krótką chwilę otwarte są oba tranzystory, co powoduje bardzo krótkie spięcie (na linii zasilania).
- Komórka SRAM'u używa tylko dwóch tranzystorów co również powoduje istotne spięcie przy zmianie wartości (w normalnym rejestrze jest osiem tranzystorów; nie licząc bramek “nie”).
- ...

Powodowanie błędów

“Glitches Attack”, “Fault Tolerance Attack”

Chwilowo zmieniając warunki działania karty powodujemy kontrolowany błąd – zmianę wartości w jednym, kilku rejestrach.
Często można też zmieniać dane przepływające pomiędzy rejestrami a pamięcią.

Techniki:

- Zmiana prędkości zegara.
- Zmiana napięcia zasilania.
- Zewnętrzne pole elektromagnetyczne.

Zabezpieczenia

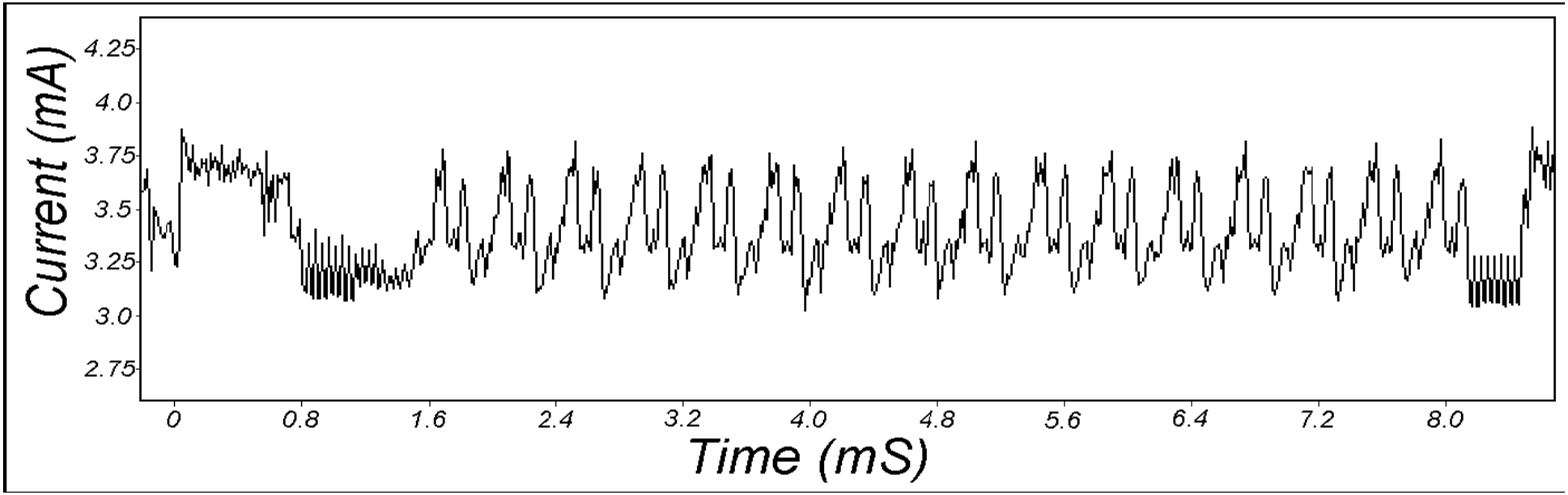
- Nieregularny zegar wewnętrzny.
- Losowa wielowątkowość (sprzętowa).
- Dobry sensor zbyt niskiej częstotliwości zegara.
- Ograniczony licznik instrukcji.
- Zdublowane wszystkie szyny, tranzystory.

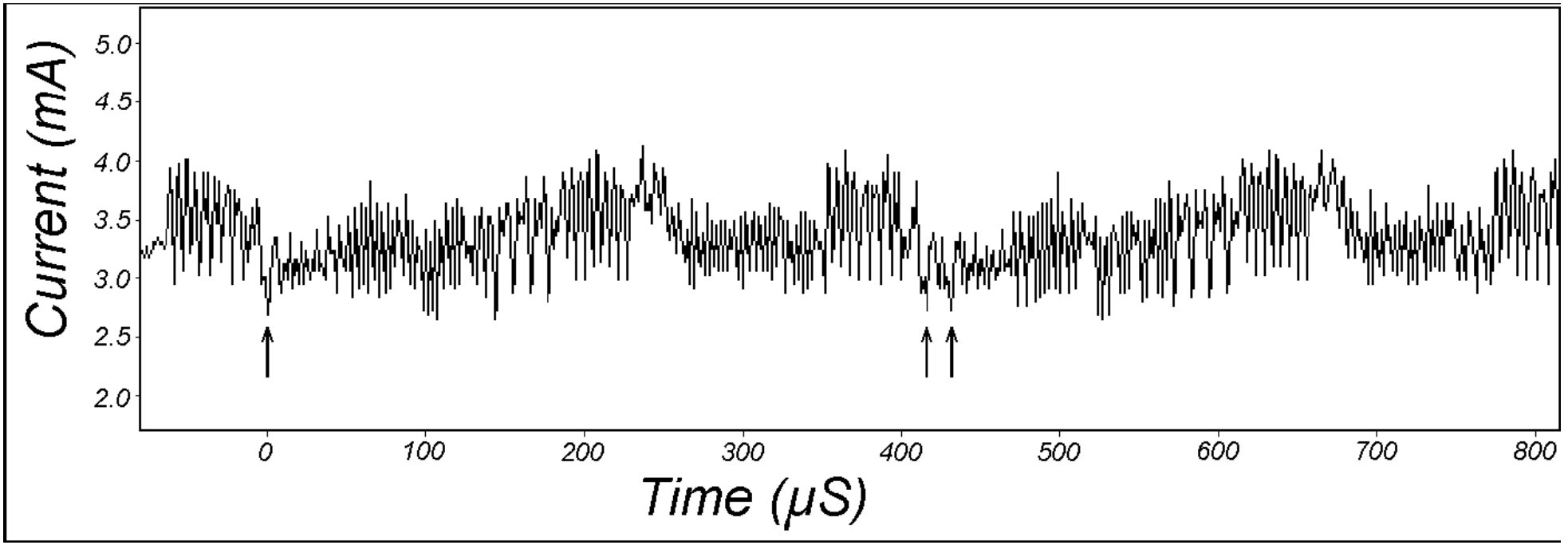
Prosta analiza natężenia

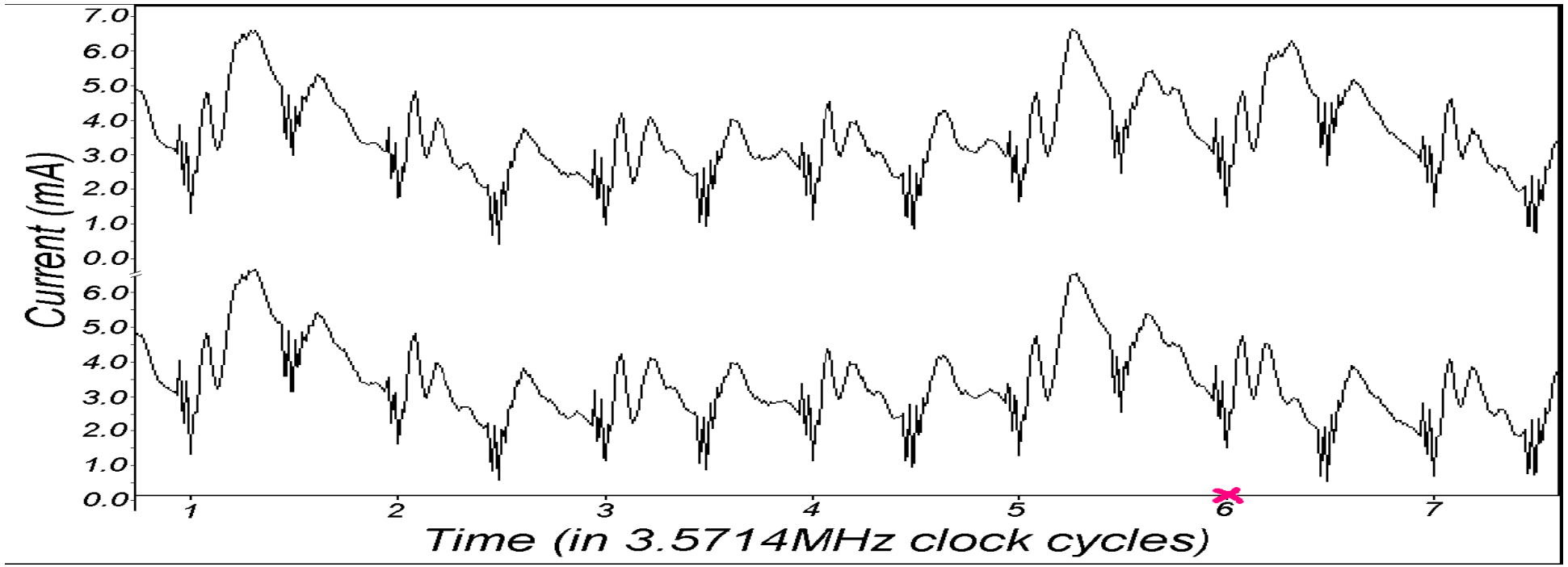
“Simple Power Analysis”, “Current Analysis”

Do zasilania podłączamy szeregowo resystorek (5-50Ω) i z dużą częstotliwością (20MHz-1GHz) mierzymy napięcie na jego zakładkach.

Różne fragmenty programu, ale także różne instrukcje, powodują różną aktywność układów mikroprocesora – różne natężenie – mogą zostać zidentyfikowane.




Current (mA)
5.0
4.5
4.0
3.5
3.0
2.5
2.0
0
100
200
300
400
500
600
700
800
Time (μS)


Current (mA)
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
1
2
3
4
5
6
7
Time (in 3.5714MHz clock cycles)

Różnicowa analiza natężenia

“Differential Power Analysis”

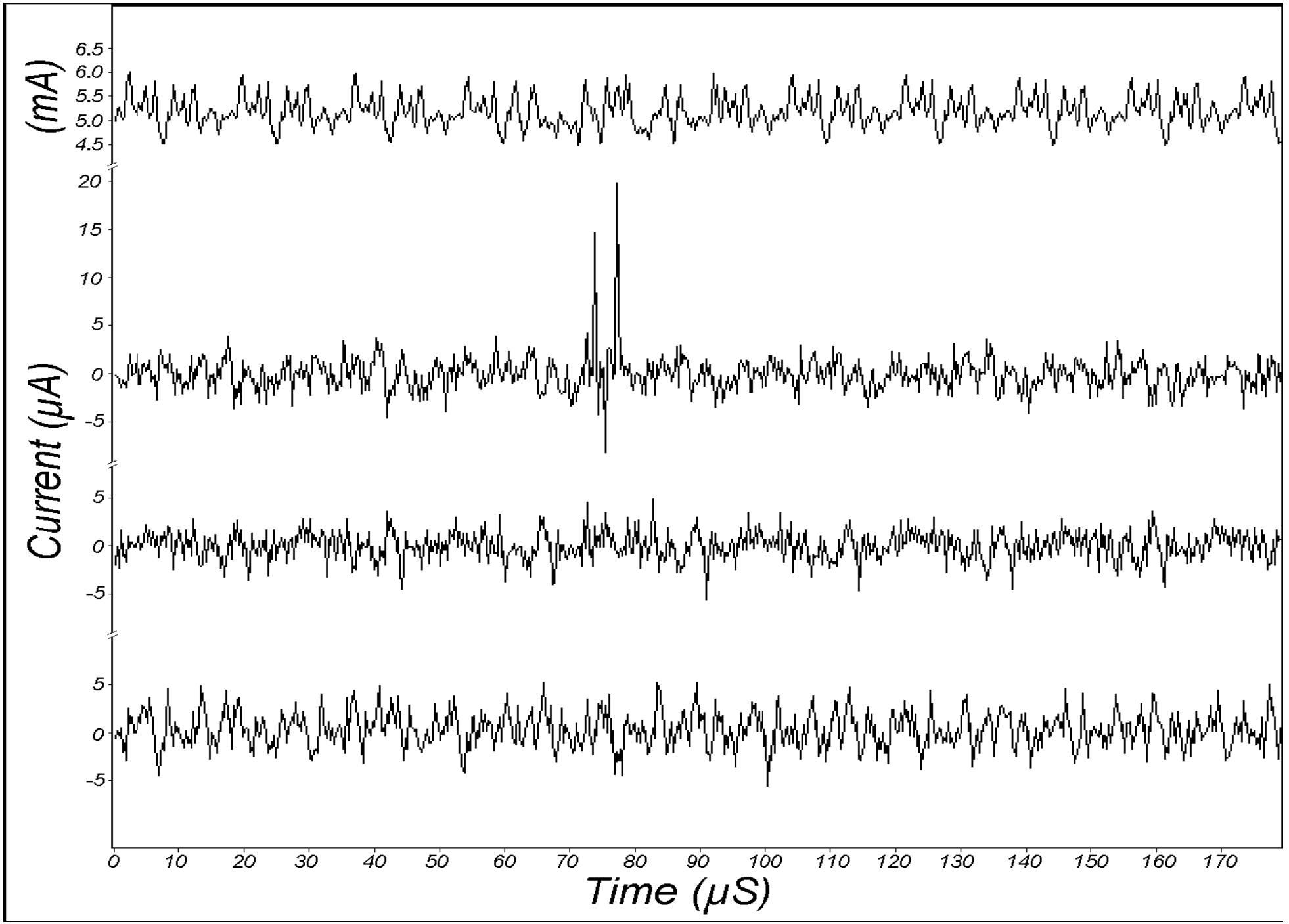
Wykorzystuje korelację pomiędzy danymi a natężeniem.

Przypuśćmy, że wykonanie pewnego kawałka kodu jest zależne od jakiegoś bitu wartości pośredniej (który możemy obliczyć zgadując kawałek klucza).

Mierzymy natężenie podczas wykonywania tego kodu dla różnych danych.
(Ciągle musi być używany ten sam klucz.)
(Np.: dla szyfrowania, podpisywania, hashowania różnych wiadomości.)

Zgadujemy klucz; obliczamy wartości wybranego bitu dla wszystkich danych.

Jeśli klucz jest zły wartości bitu będą nieskorelowane z wartościami napięcia jeśli jest dobry jest spora szansa na dużą korelację.


(mA)
6.5
6.0
5.5
5.0
4.5
Current (µA)
20
15
10
5
0
-5
5
0
-5
5
0
-5
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
Time (µS)

Zapobieganie SPA i DPA

- Redukcja sygnału:
  - Ścieżka wykonania niezależna od sekretów (w szczególności kluczy:)
  - Nieużywanie procedur tworzących sekretne wartości pośrednie.
  - Instrukcje procesora, mikrokod, działające podobnie niezależnie od wartości argumentów; wykorzystywanie takich instrukcji z których mniej wycieka.
  - Oślepianie.
  - Realizacje sprzętowe zamiast oprogramowania.
  - Fizyczne osłanianie.

- Dodawanie szumu.

- Odpowiednie projektowanie algorytmów – z założeniem, że dużo, (trzeba ustalić konkretnie ile) może wyciec.

Bibliografia

- *“Breaking Up Is Hard To Do: Modeling Security Threats for Smart Cards”*, Bruce Shneier, Adam Shostack
- *“Design Principles for Tamper-Resistant Smartcard Processors”*, Oliver Kömmerling, Markus G. Kuhn
- *“Differential Power Analysis”*, Paul Kocher, Joshua Jae i Benjamin Jun
- *“Overview about Attacks on Smart Cards”*, Wolfgang Rankl

maj 2006

Wojtek.Ruszczewski@gmail.com