Analiza możliwości sprzętowej implementacji szyfru blokowego opartego o algorytm Anubis w sposób uodparniający na ataki kanałem bocznym

sierż. pchor. Damian Krata

Wojskowa Akademia Techniczna

Warszawa
15.04.2019

Algorytm Anubis

Warstwa nieliniowa γ składa się z równolegle działających skrzynek podstawieniowych $S : \mathrm{GF}(2^8) \to \mathrm{GF}(2^8)$, $x \mapsto S[x]$, które realizują podstawienie każdego bajtu z osobna:

$$\gamma(a) = b \Leftrightarrow b_{ij} = S[a_{ij}],\ 0 \leqslant i \leqslant N-1,\ 0 \leqslant j \leqslant 3.$$

Sbox został wybrany pseudo - losowo tak, aby zapewnić warunek: $S[S[x]] = x$ dla każdego $x \in \mathrm{GF}(2^8)$.

Transpozycja τ

Mapowanie $\tau : \mathcal{M}_{4\times4}[\mathrm{GF}(2^8)] \rightarrow \mathcal{M}_{4\times4}[\mathrm{GF}(2^8)]$ transponuje elementy macierzy stanu:

$$\tau(a) = b \Leftrightarrow b = a^t \Leftrightarrow b_{ij} = a_{ji},\ 0 \leqslant i, j \leqslant 3.$$

Transpozycja jest inwolucją.

Warstwa dyfuzji θ

$$\theta(a) = b \Leftrightarrow b = a \cdot H,$$

gdzie $H = \mathrm{had}(\text{'01'}, \text{'02'}, \text{'04'}, \text{'06'})$, tzn.

$$H = \begin{bmatrix} \text{'01'} & \text{'02'} & \text{'04'} & \text{'06'} \\ \text{'02'} & \text{'01'} & \text{'06'} & \text{'04'} \\ \text{'04'} & \text{'06'} & \text{'01'} & \text{'02'} \\ \text{'06'} & \text{'04'} & \text{'02'} & \text{'01'} \end{bmatrix}$$

Dodanie klucza $\sigma[k]$

$$\sigma[k](a) = b \Leftrightarrow b_{ij} = a_{ij} \oplus k_{ij},\ 0 \leqslant i \leqslant N-1,\ 0 \leqslant j \leqslant 3.$$

Cykliczna permutacja π

Permutacja $\pi : \mathcal{M}_{N\times 4}[\mathrm{GF}(2^8)] \rightarrow \mathcal{M}_{N\times 4}[\mathrm{GF}(2^8)]$, $4 \leqslant N \leqslant 10$, cyklicznie przesuwa każdą kolumnę oddzielnie do dołu, w ten sposób, że kolumna j jest przesuwana o j pozycji:

$$\pi(a) = b \Leftrightarrow b_{ij} = a_{(i-j) \bmod N, j},\ 0 \leqslant i \leqslant N-1,\ 0 \leqslant j \leqslant 3.$$

Ekstrakcja klucza ω

$$\omega(a) = b \Leftrightarrow b = V \cdot a,$$

gdzie $V = \mathrm{vdm}_N(\text{‘01’}, \text{‘02’}, \text{‘06’}, \text{‘08’})$, tzn.

$$V = \begin{bmatrix} \text{‘01’} & \text{‘01’} & \text{‘01’} & \dots & \text{‘01’} \\ \text{‘01’} & \text{‘02’} & \text{‘02’}^2 & \dots & \text{‘02’}^{N-1} \\ \text{‘01’} & \text{‘06’} & \text{‘06’}^2 & \dots & \text{‘06’}^{N-1} \\ \text{‘01’} & \text{‘08’} & \text{‘08’}^2 & \dots & \text{‘08’}^{N-1} \end{bmatrix},$$

Schemat klucza

Schemat klucza (ang. *key schedule*) rozszerza klucz $K \in \mathrm{GF}(2^8)^{4N}$, $4 \leqslant N \leqslant 10$, na klucze rund $K^0, \ldots, K^R$, gdzie $K^r \in \mathcal{M}_{4\times 4}[\mathrm{GF}(2^8)]$:

$$\begin{aligned} \kappa^0 &= \mu(K), \\ \kappa^r &= (\sigma[c^r] \circ \theta \circ \pi \circ \gamma)(\kappa^{r-1}), \quad r > 0, \\ K^r &= (\tau \circ \omega \circ \gamma)(\kappa^r), \quad 0 \leqslant r \leqslant R; \end{aligned}$$

Przekształcenie $\psi[c^r] \equiv \sigma[c^r] \circ \theta \circ \pi \circ \gamma$ nazywane jest funkcją rozwoju r-tego klucza rundy podczas gdy $\phi \equiv \tau \circ \omega \circ \gamma$ nazywane jest funkcją wyboru klucza. W ogólności w fazie obliczeń, wyznaczanie klucza rundy sprowadza się do wykonania funkcji ϕ na kluczu poddawanym obliczeniom za pomocą funkcji ψ.

Matematyczny opis algorytmu

Dla klucza $K \in \mathrm{GF}(2^8)^{4N}$, Anubis może być zdefiniowany jako przekształcenie $\textsc{Anubis}[K] : \mathrm{GF}(2^8)^{16} \to \mathrm{GF}(2^8)^{16}$ zadane przez

$$\textsc{Anubis}[K] \equiv \mu^{-1} \circ \alpha_R[K^0, \ldots, K^R] \circ \mu,$$

gdzie

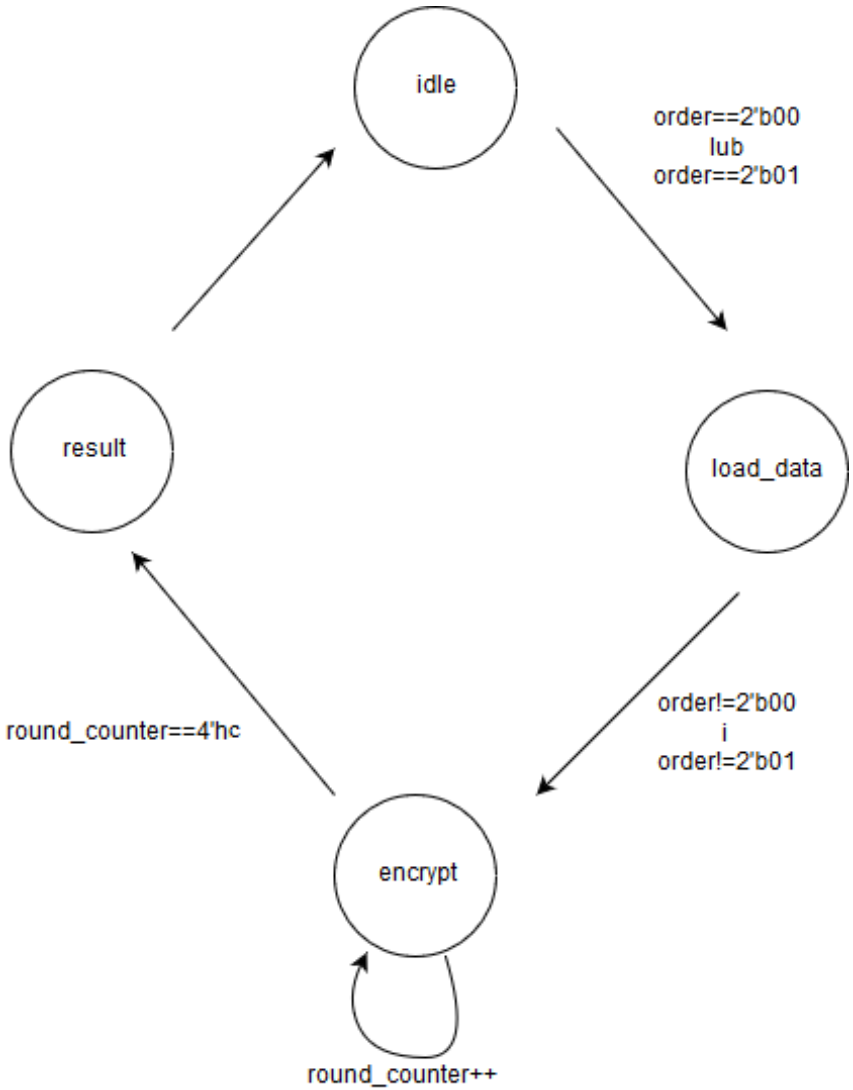
$$\alpha_R[K^0, \ldots, K^R] = \sigma[K^R] \circ \tau \circ \gamma \circ \left(\bigcirc_{1}^{r=R-1} \sigma[K^r] \circ \theta \circ \tau \circ \gamma \right) \circ \sigma[K^0].$$

Standardowa liczba rund dla algorytmu R określona jest jako $R = 8 + N$ dla $32N$-bitowego klucza, $4 \leqslant N \leqslant 10$. Przekształcenie $\rho[K^r] \equiv \sigma[K^r] \circ \theta \circ \tau \circ \gamma$ nazywane jest *funkcją rundy*, natomiast $\rho'[K^R] \equiv \sigma[K^r] \circ \tau \circ \gamma$ pozbawione θ określane jest jako *funkcja ostatniej rundy*.

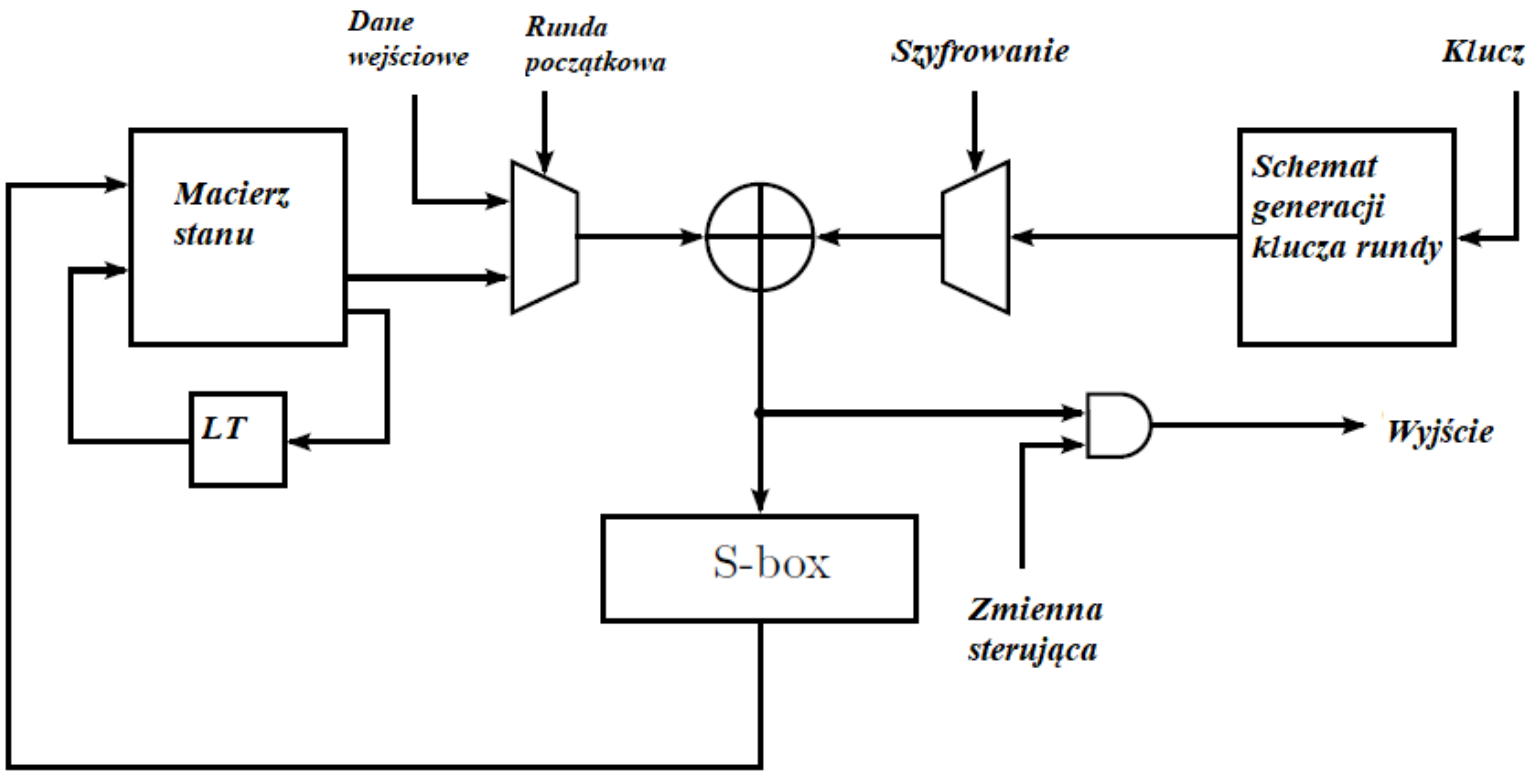
Porównanie Anubis i AES

	AES	Anubis
Rozmiar bloku	128	128
Rozmiar klucza	128, 192, 256	128, 160, 192, 224, 256, 288, 320
Liczba rund	10, 12, 14	12, 13, 14, 15, 16, 17, 18
Schemat tworzenia kluczy	algorytm dedykowany *a priori*	funkcje rozwijania i wyboru klucza
Wielomian redukujący $GF(2^8)$	$x^8 + x^4 + x^3 + x + 1$ $(0x11B)$	$x^8 + x^4 + x^3 + x^2 + 1$ $(0x11D)$
Pochodzenie Sbox'a	odwrotność w ciele $GF(2^8)$ i przekształcenie afiniczne	losowo wybrana inwolucja
Pochodzenie stałych rundy	wielomiany x^i nad $GF(2^8)$	kolejne wejścia do Sbox'a

Maszyna stanów



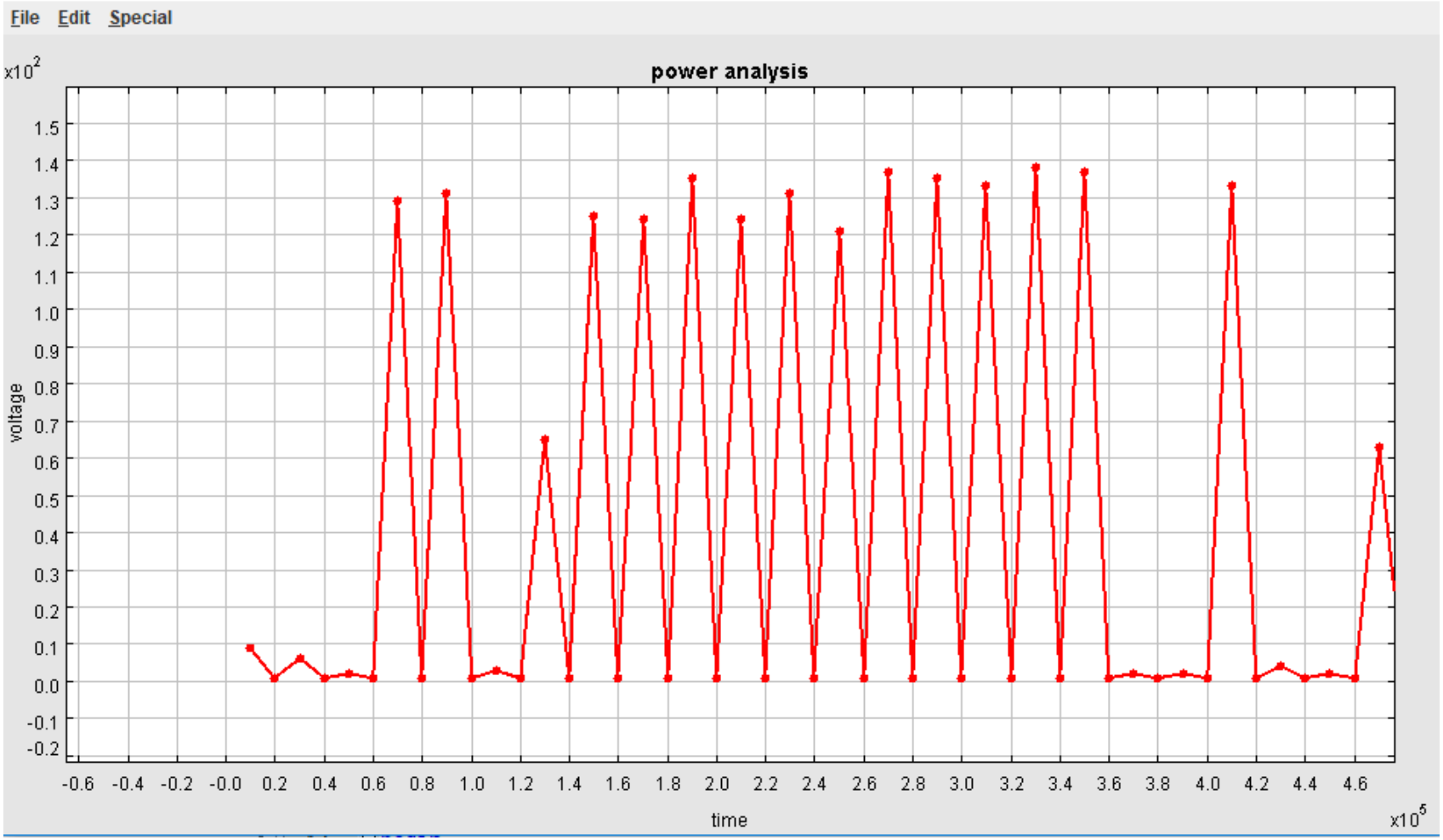
Schemat implementacji



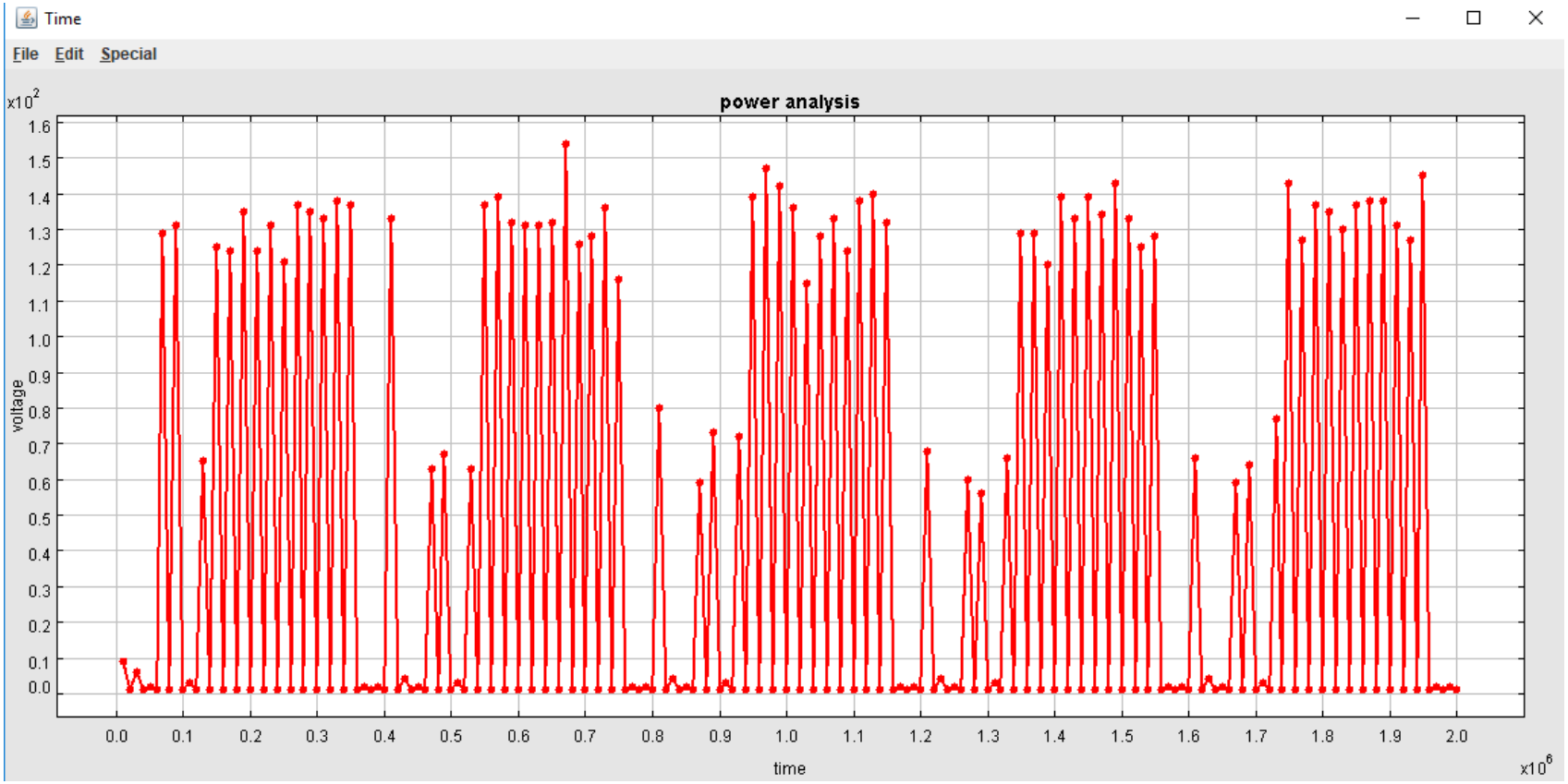
Użyte narzędzia

- testbench,
- python do generacji wektorów testowych,
- dane z symulacji do listingu,
- analiza danych za pomocą programu napisanego w Java.

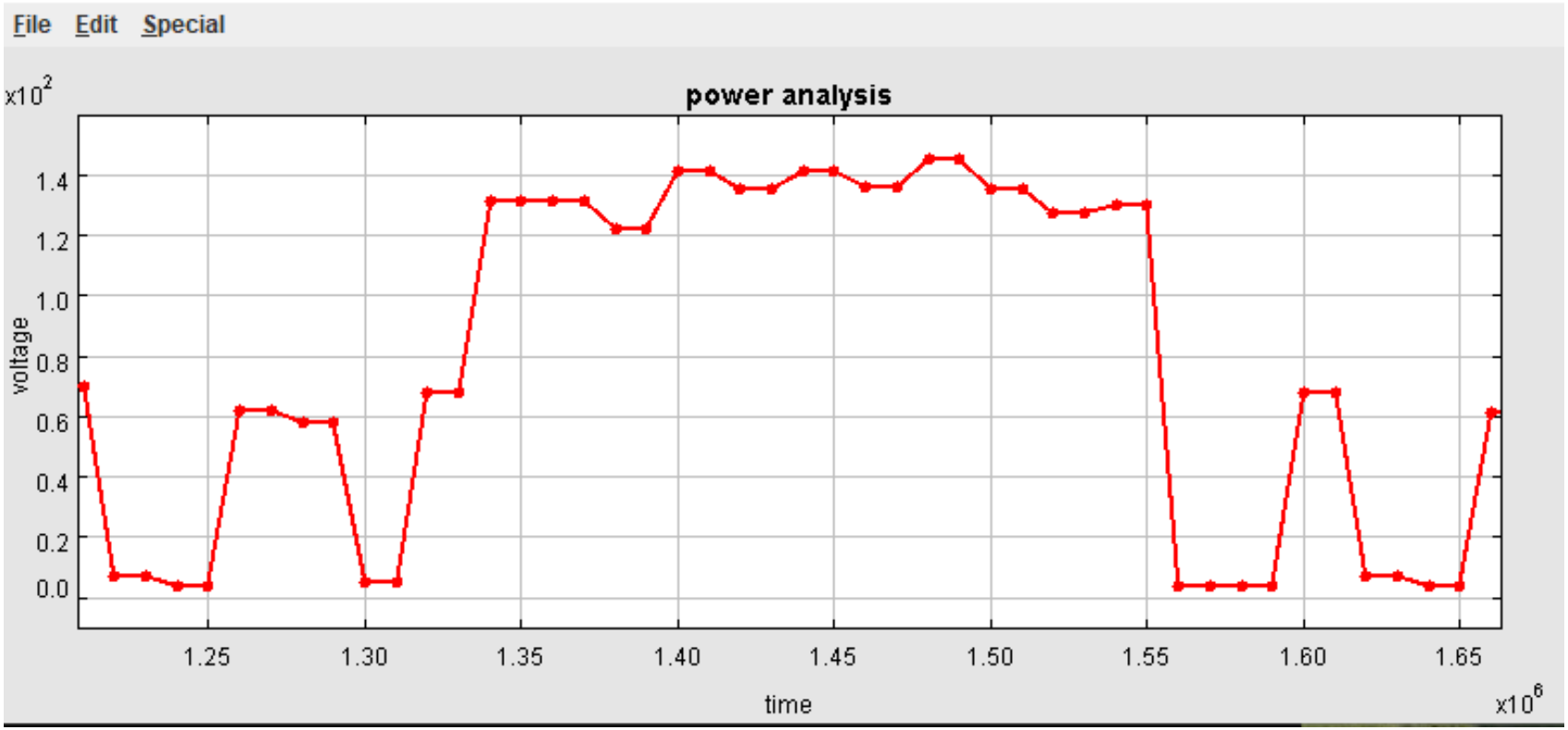
Wynik dla symulacji referencyjnej



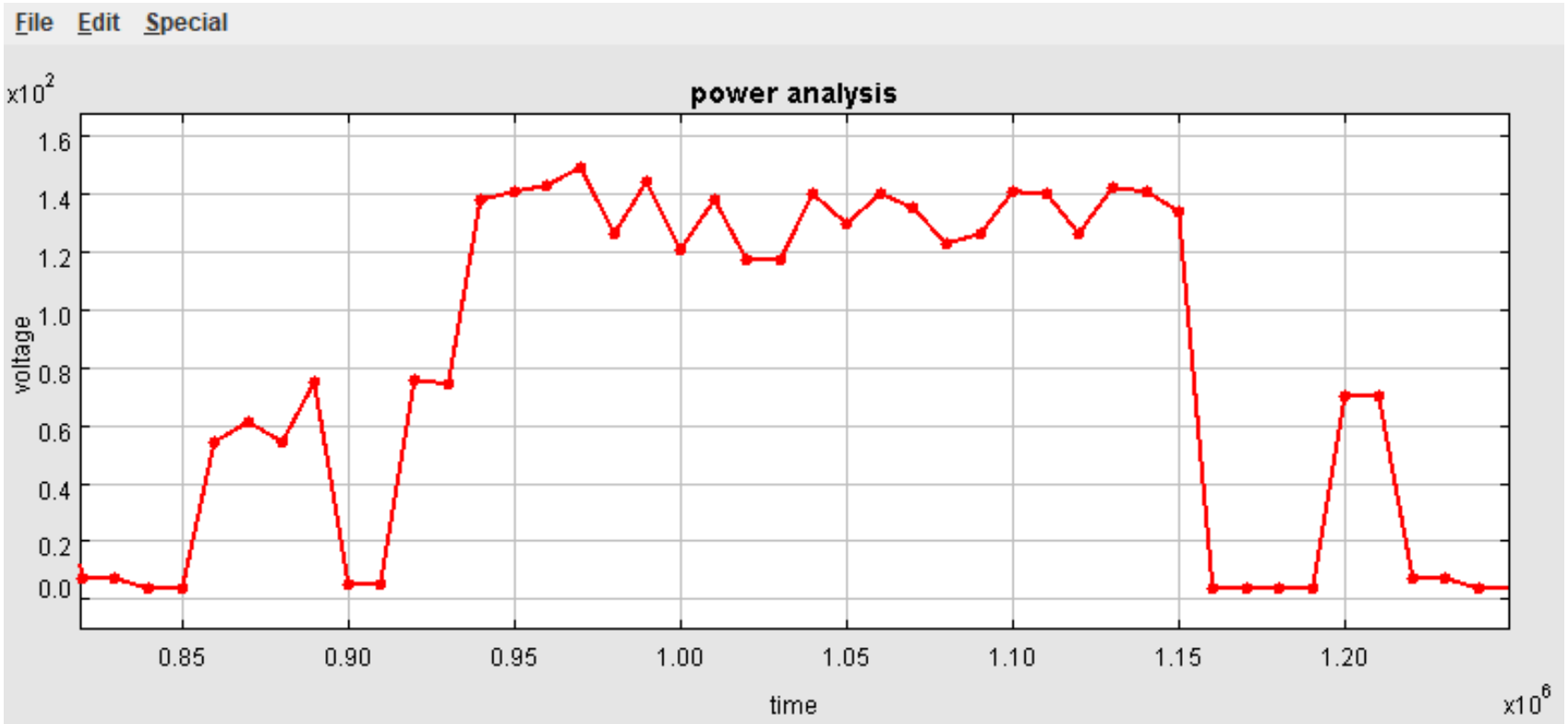
Wynik dla symulacji referencyjnej



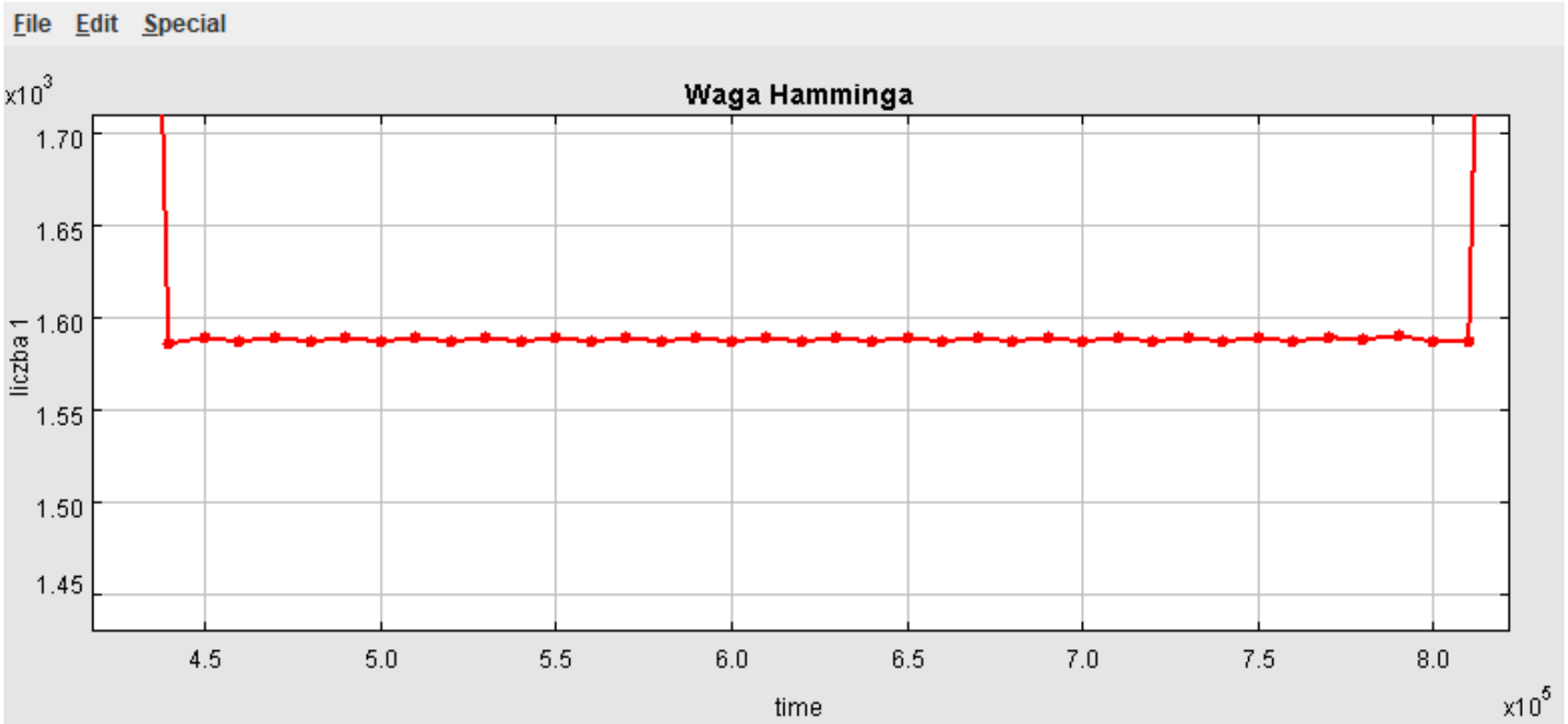
Druga implementacja - dodanie zbocza opadającego - 1



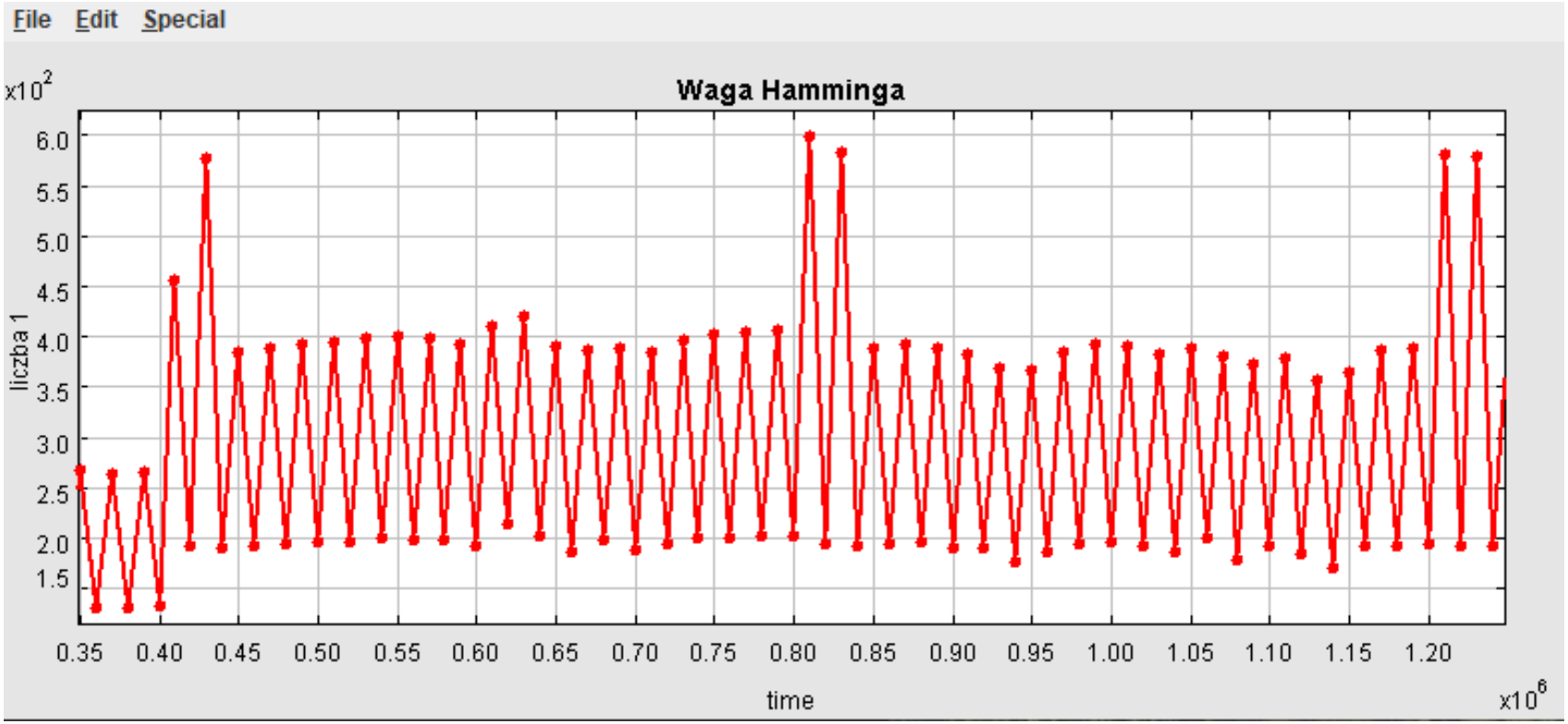
Druga implementacja - dodanie zbocza opadającego - 2



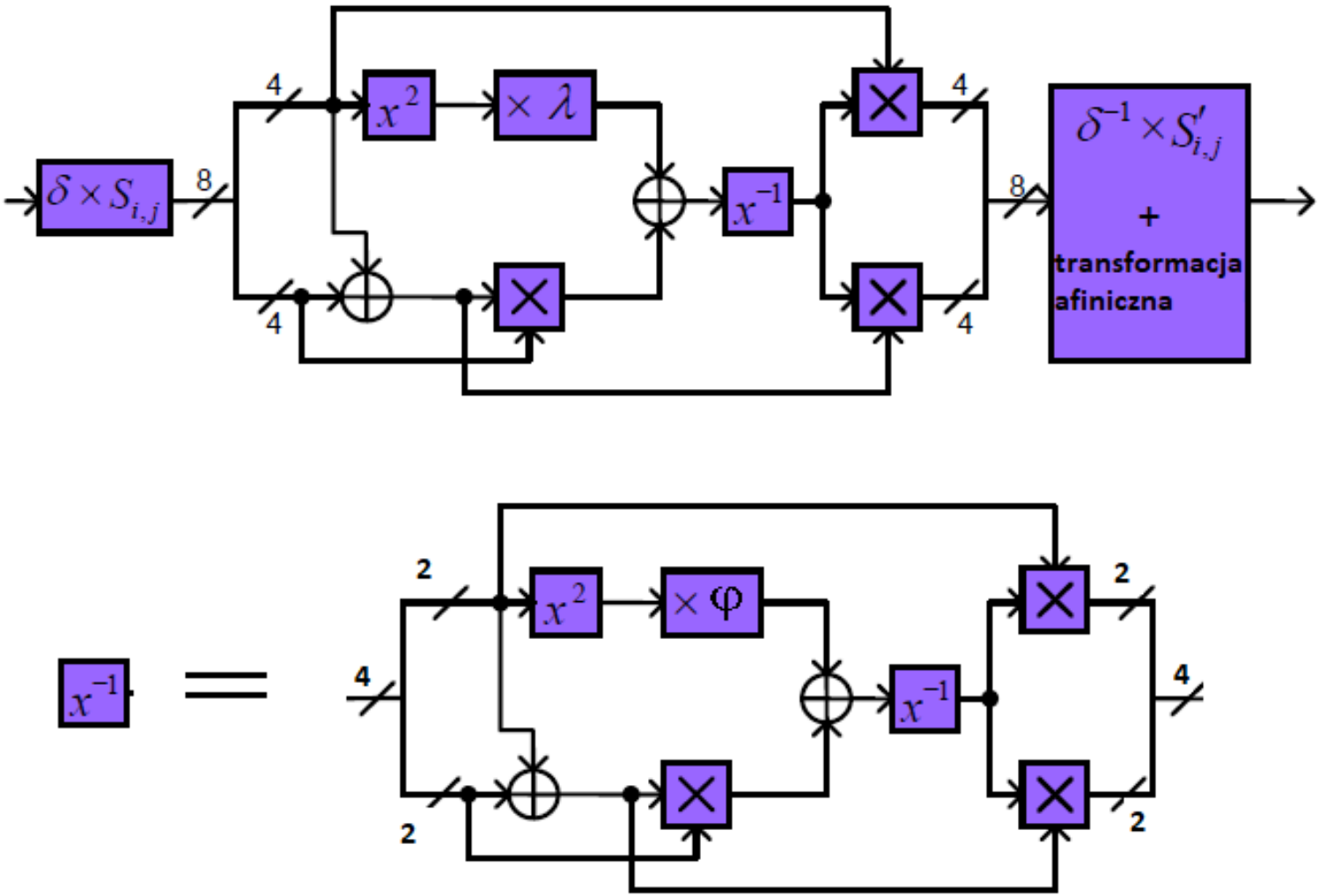
Trzecia implementacja - dodanie rejestrów



W porównaniu do referencyjnej



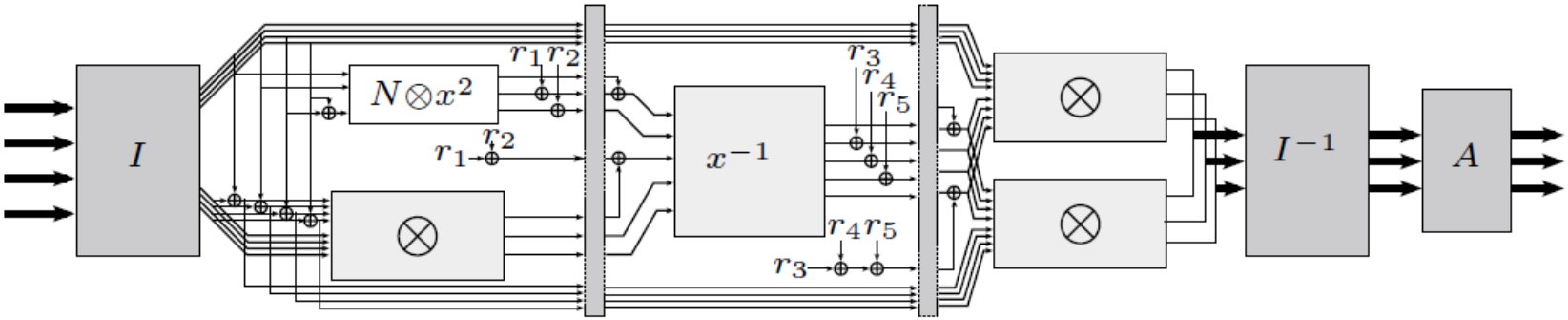
Implementacja czwarta - podmiana skrzynki na AES



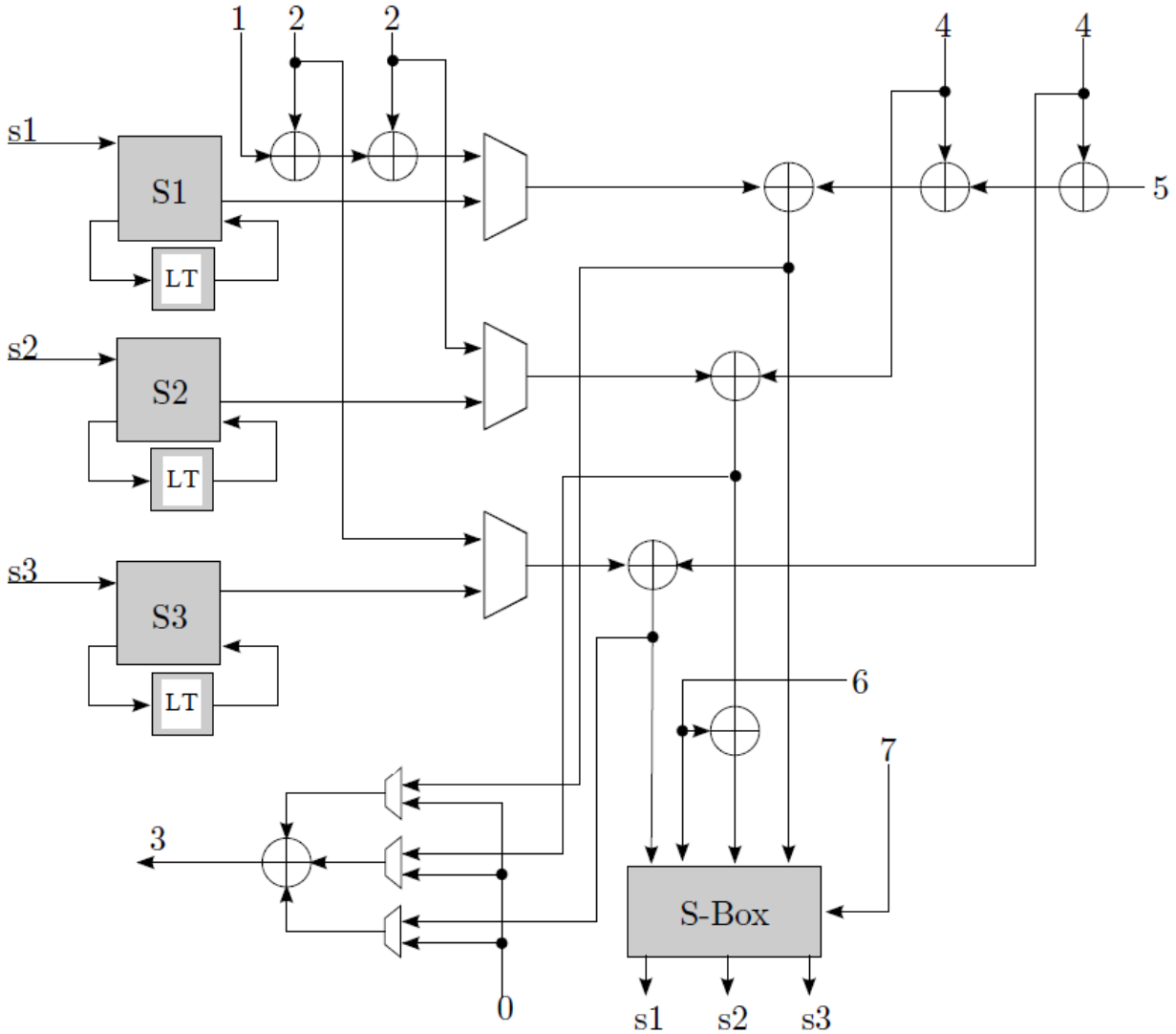
Schemat dzielenia sekretu

$$
\begin{aligned}
F &= XY, \\
F &= F_1 \oplus F_2 \oplus F_3, \\
X &= X_1 \oplus X_2 \oplus X_3 \oplus X_4, \\
Y &= Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4, \\
F_1 &= (X_2 \oplus X_3 \oplus X_4)(Y_2 \oplus Y_3) \oplus Y_4, \\
F_2 &= ((X_1 \oplus X_3)(Y_1 \oplus Y_4)) \oplus X_1 Y_3 \oplus X_4, \\
F_3 &= ((X_2 \oplus X_4)(Y_1 \oplus Y_4)) \oplus X_1 Y_2 \oplus X_4 \oplus Y_4.
\end{aligned}
$$

Zastosowanie schematu do skrzynki podstawieniowej



Schemat uodpornionego algorytmu



Wydajność implementacji z pierwszej grupy

	Implementacja I	Implementacja II	Implementacja III
$85°C$	72,31 MHz	70,72 MHz	72,35 MHz
$0°C$	73,28 MHz	71,42 MHz	73,41 MHz

Tabela: Maksymalne częstotliwości taktowania zegara dla implementacji z pierwszej grupy.

	Implementacja I	Implementacja II	Implementacja III
ALM	2652	5127	5472
reg	524	1048	1300

Tabela: Zajętość zasobów dla poszczególnych implementacji z pierwszej grupy.

Wydajność implementacji z drugiej grupy

	Implementacja IV	Implementacja V	Implementacja VI	Implementacja VII
$85°C$	73,36 MHz	55,32 MHz	56,48 MHz	25,59 MHz
$0°C$	74,16 MHz	56,01 MHz	57,2 MHz	25,98 MHz

Tabela: Maksymalne częstotliwości taktowania zegara dla implementacji z drugiej grupy.

	Implementacja IV	Implementacja V	Implementacja VI	Implementacja VII
ALM	2624	4786	9970	19360
reg	524	524	780	1680

Tabela: Zajętość zasobów dla poszczególnych implementacji z drugiej grupy.

Koniec

Dziękuję za uwagę!