attack-aware self-logic-transformation cryptographic algorithm
Theoretical Design and Possibility

Abstract

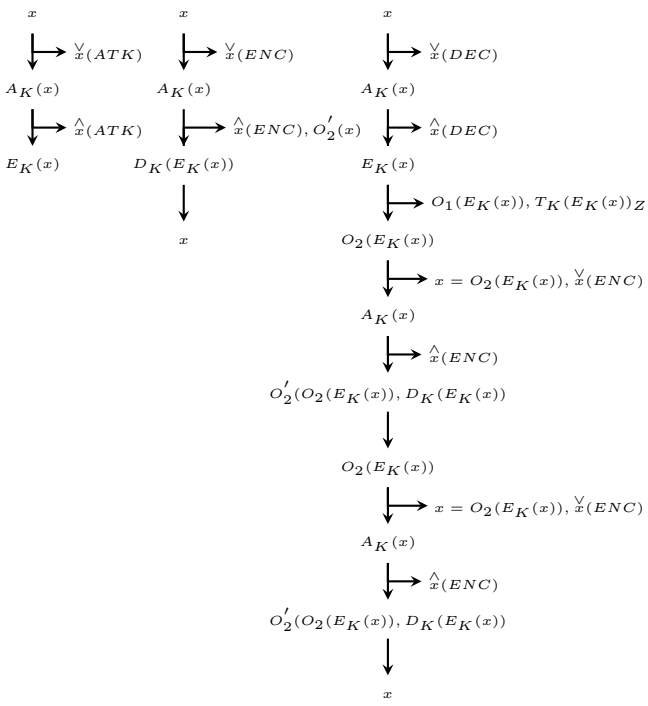
this part introduces theoretical design of this algorithm based on deep learning for cryptographic purpose and its possibility before implement this algorithm. theoretical design covers structure and logic for this algorithm and some experiments and equation for this algorithm have been included to see the possibility of this algorithm based on deep learning for data encryption and decryption. this algorithm continuously feeds data and detects and determines whether attack exists on the data and the determination affects and changes its structure and the information of each units in the structure. expected effects of this algorithm are hiding cryptographic operation and cryptographic key into internals in the structure and veiling data and its status and the result from cryptographic operation executed on data to data. this algorithm mainly consists of two phases, attack-aware phase determining that attack on data exists and self-logic-transformation phase executing cryptographic operation on data.

1 Definition

this section introduces some notations and logical flow of this algorithm and their logical relations under conditions made for the algorithm.

when x = input data, K = cycle number, $A_K(x)$ = attack-aware function, $E_K(x)$ = encrypt function, $D_K(x)$ = decrypt function, $T_K(x)_N$ = trick function, M = maximum value in the number space of N, $O_1(x)$ = middle output function, $O_2(x)$ = output function, $O_2^{'}(x)$ = reverse calculation of $O_2(x)$, S = status of data, ATK = status of data attacked, ENC = status of data encrypted, DEC = status of data decrypted, $\overset{\vee}{x}(S)$ = data x with S not determined, $\hat{x}(S)$ = data x with S determined.

examples of the logical flow for ATK, ENC and DEC have been described with the above notations.



their logical relations for this algorithm have been made.

$E_K(x) = E_K(x) \neq E_K(x)$
$T_K(E_K(x))_Z = T_K(E_K(x))_Z \neq T_K(E_K(x))_Z$
$O_1(x) = O_1(x) \neq O_1(x)$
$O_2(x) = O_2(x) \neq O_2(x), O_2^{'}(O_2(x)) = x$
$D_{K_1}(E_{K_1}(X)) = D_{K_2}(E_{K_2}(X)) = x, D_{K_1}(E_{K_2}(x)) = x$
$A_K(\overset{\vee}{x}(ATK)) = \hat{x}(ATK)$
$A_K(\overset{\vee}{x}(ENC)) = \hat{x}(ENC), A_K(\overset{\vee}{x}(DEC)) = \hat{x}(DEC)$
$A_{K_1}(\overset{\vee}{x}(S)) = A_{K_2}(\overset{\vee}{x}(S)) = \hat{x}(S)$

2 Equation

the equations from (1) to (4) have been used for this algorithm.

$$O_1(x) = sign(E_K(x)) \tag{1}$$
$$T_K(x)_N = (O_1(x) - E_K(x)) mod(M) \tag{2}$$
$$O_2(x) = O_1(x) + T(x)/2 \tag{3}$$
$$k = \frac{\alpha x + (\alpha - 1 - \frac{\alpha}{y})y}{\alpha x y} \tag{4}$$

3 Experiment

this section explains some experiments for deep learning based data encryption and decryption and some results from experiments.

x	ENC_1	DEC_1	ENC_2	DEC_2
48	40.14215	49.69979	96.5069	138.5262
97	226.3464	1.979933	84.02759	93.61181

x	ENC	O_1	T	O_2	O_1 from O_2
48	96.5069	6.24853	164.742	88.6193	6
97	84.0276	29.5091	200.482	129.75	29

a	x	y	$\lfloor x \rfloor$	$\lceil y \rceil$
1	96	89	96	89
1.000001	130	84	131	84

4 Determinism

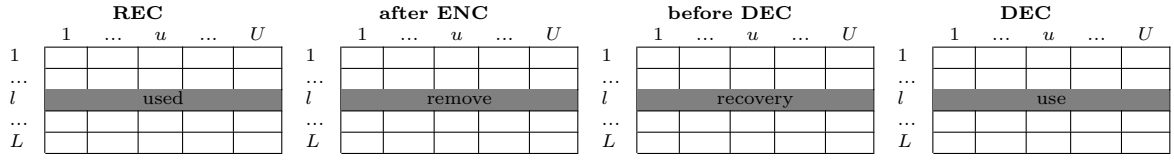
this section introduces the problems related with determinism for this algorithm.

5 Methodology

this section introduces a method for this algorithm.

6 Structure

this section introduces based on deep learning for this algorithm.


REC
after ENC
before DEC
DEC
1
u
U
l
L
used
remove
recovery
use