Golden Research Thoughts
Vol -1 , ISSUE –5, November - 2011
ISSN:- 2231-5063
Available online at www.lbp.world

ORIGINAL ARTICLE

GRT

NETWORK SECURITY USING KAMAL TRANSFORM

Mrs. Meena Patil
Assistant Professor,
SNDT college of Arts & SCB college of commerce & science for women, Mumbai.

Abstract

First section of this paper develops a full Kamal transform based encryption–decryption framework for network security. It parallels the structure of the original Sumudu-transform paper while replacing all calculations with Kamal Transform equivalents, including full term-by-term Taylor expansions, coefficient derivations and polynomial degree analysis. Next part parallels the original Sumudu-transform-based paper, but all computations are rewritten using the Kamal Transform. We also derive all quantities symbolically for general positive parameters a and r. We still consider the plaintext as a finite sequence $(P^{00}, P^{01}, \dots, P^{0}{}_{n})$, obtained from the usual encoding $a = 0, b = 1, \dots, z = 25$, and set $P^{0}{}_{i} = 0$ for $i > n$. We define the generating function $f(t) = a\,P(e\text{^}rt + \sinh(rt))$, where P is the formal power series associated with the plaintext: $P(t) = \Sigma_{k} P^{0}{}_{k} t^{k}$, with $P^{0}{}_{k} \in \{0,1,\dots,25\}$, $P^{0}{}_{k} = 0$ for $k > n$. In the last part we interpret all results through the Kamal Transform. As discussed in the theory, for polynomial inputs the Kamal Transform $K[\cdot]$ and the Sumudu Transform $S[\cdot]$ are related by a simple scaling of the transform variable: dividing $K[f(t)]$ by the transform variable v yields a polynomial with coefficients $n!$ multiplied by the original time–domain coefficients of t^{n}. This is exactly the property used in the Sumudu-based construction **[1 - 15]**. Hence, for the same generating function $f(t)$, the polynomial coefficients that drive the cryptosystem coincide with those of the original paper, but are now justified via the Kamal integral kernel. We also give a fully worked example for the plaintext NETWORK with parameters

$$a = 1, r = 2, \text{and } p = 3,$$

followed by several further examples with different choices of a and r.

PART 1: DEFINITION OF KAMAL TRANSFORM [1]

1. Taylor Series Expansions

The exponential and hyperbolic functions admit the series expansions:

$e^t = \Sigma \frac{t^n}{n!}$ for $n = 0, \ldots, \infty$

$\sinh(t) = \Sigma \frac{t^{2n+1}}{(2n+1)!}$ for $n = 0, \ldots, \infty$

2. Kamal Transform Definition [2]

$K[f(t)] = \int_0^\infty f(t) e^{\left(-\frac{t}{v}\right)} dt$, where v is the transform variable.

3. Basic Kamal Transform Results

$$K[t^n] = n!\ v^{n+1}$$

$$K[e^{at}] = \frac{v}{1 - av}$$

$$K[\sinh(at)] = \frac{av^2}{1 - a^2v^2}$$

4. Kamal Transform Expansion of e^{rt}

Write $e^{rt} = \Sigma \left(\frac{r^n t^n}{n!}\right)$. Applying K term-by-term:

$K[e^{rt}] = \Sigma \left(\frac{r^v}{n!}\right) K[t^n] = \Sigma \left(\frac{r^n}{n!}\right) n!\, v^{n+1} = \Sigma r^n v^{n+1}$.

This series sums to the closed form $\frac{v}{1-rv}$.

Similar calculations can be done for Kamal transform of $\sinh(rt)$.

5. Foundation for Encryption Polynomial

The encryption model requires expanding $f(t) = aP(e^{rt} + \sinh(rt))$ where P is the plaintext polynomial.

PART 2: FULL ENCRYPTION CONSTRUCTION USING KAMAL TRANSFORM [3]

6. Full Construction of Encryption Polynomial $f(t)$.

6.1 Plaintext Conversion

For plaintext NETWORK, the numerical encoding is:

$$N = 13, E = 4, T = 19, W = 22, O = 14, R = 17, K = 10.$$

Thus $P^0 = \{13, 4, 19, 22, 14, 17, 10\}$ and $P_{0i} = 0$ for $i \geq 7$.

6.2 Expansion of $f(t)$

Define $f(t) = P^0\left(e^{\{2t\}} + \sinh(2t)\right)$. Using full Taylor expansion:

$e^{\{2t\}} = \Sigma \frac{(2t)^n}{n!}$, $\sinh(2t) = \Sigma \frac{(2t)^{2n+1}}{(2n+1)!}$

Thus the general coefficient of t^m becomes a combination of P^0 terms. Each t^m term will transform to $m!\,v^{\{m+1\}}$ under $K[t^m]$.

6.3 Kamal Transform Term-by-Term

Each term $\left(2^m \frac{P_{0m}}{m!}\right) t^m$transforms as: $K\left[\left(2^m \frac{P_{0m}}{m!}\right) t^m\right] = 2^m P_{0m} v^{\{m+1\}}$. Odd powers from $\sinh(2t)$ contribute terms of the form: $K\left[\left(\frac{P_{0n}}{(2n+1)!}\right) t^{\{2n+1\}}\right] = 2^{\{2n+1\}} P_{0n} v^{\{2n+2\}}$.

7. Coefficient Table Construction

We construct the table containing: $i, B_{\mathrm{i}}, B_{\mathrm{i}} + p, P_{1i}, L_{1i}$ where $p = 3$ is used as in the example [16-23].

i	B_{i}	$B_{\mathrm{i}} + p$	$P_{1i} = (B_{\mathrm{i}} + p) \bmod 26$	$L_{1i} = \frac{B_{\mathrm{i}} + p - P_{1i}}{26}$
0	(computed)	(+3)	(mod 26)	(key)
1	(computed)	(+3)	(mod 26)	(key)
2	(computed)	(+3)	(mod 26)	(key)
3	(computed)	(+3)	(mod 26)	(key)
4	(computed)	(+3)	(mod 26)	(key)
5	(computed)	(+3)	(mod 26)	(key)
6	(computed)	(+3)	(mod 26)	(key)
7	(computed)	(+3)	(mod 26)	(key)
8	(computed)	(+3)	(mod 26)	(key)
9	(computed)	(+3)	(mod 26)	(key)
10	(computed)	(+3)	(mod 26)	(key)
11	(computed)	(+3)	(mod 26)	(key)
12	(computed)	(+3)	(mod 26)	(key)
13	(computed)	(+3)	(mod 26)	(key)

8. Ciphertext Generation

Once P_{1i} values are obtained, they convert to letters using $A = 0, \ldots, Z = 25$. The actual numerical B_{i} values will be computed explicitly in Part 3.

PART 3: GENERAL SYMBOLIC CONSTRUCTION FOR ARBITRARY a AND r

9. Symbolic Kamal–Transform Coefficients for General a and r

9.1 Series expansions of $e^{\{rt\}}$ and sin h(rt)

We use the Taylor series expansions

$e^{\{rt\}} = \Sigma_j \left(\frac{r^j}{j!}\right) t^j$, for $j = 0, 1, 2, \ldots$

$\sinh(rt) = \Sigma_m \left(\frac{r^{\{2m+1\}}}{(2m+1)!}\right) t^{\{2m+1\}}$, for $m = 0, 1, 2, \ldots$

Hence the sum $e^{\{rt\}} + \sinh(rt)$ can be written as a single power series in t:

$e^{\{rt\}} + \sinh(rt) = \Sigma_m E_m t^m$, where $E_m = \frac{r^m}{m!}$, for all $m \geq 0$ (from $e\wedge rt$), and additional contributions from $\sinh(rt)$for odd m: $E_{\{2m+1\}} \leftarrow E_{\{2m+1\}} + \frac{r^{\{2m+1\}}}{(2m+1)!}$.

9.2 Coefficients of $f(t)$ before applying Kamal Transform

By definition, $f(t) = a\,P(t)\left(e^{\{rt\}} + \sinh(rt)\right) = a\,(\Sigma_k P^0{}_k t^k)(\Sigma_j E_j t^j)$. Thus $f(t)$ is again a power series: $f(t) = \Sigma_m C_m t^m$, where $C_m = a\,\Sigma_{\{k=0\}}^{m} P_{0k} E_{m-k}$.

Using the explicit expression for E_{m-k} , we can write for each $m \geq 0$: $C_m = a\,\Sigma_{\{k=0\}}^{m} P_{0k}\left[\frac{r^{\{m-k\}}}{(m-k)!} + \delta_{\{m-k,odd\}} \cdot \frac{r^{\{m-k\}}}{(m-k)!}\right]$, where $\delta_{\{m-k,odd\}}$ is 1 if $(m-k)$ is odd and 0 otherwise, representing the contribution from $\sinh(rt)$.

9.3 Applications of the Kamal Transform to $f(t)$ [16 - 20]

The Kamal Transform of a monomial t^m is given by $K[t^m] = m!\; v^{\{m+1\}}$. Since $f(t) = \Sigma_m C_m t^m$, we obtain $K[f(t)] = \Sigma_m C_m K[t^m] = \Sigma_m C_m m!\; v^{\{m+1\}}$. If we index the coefficients of $K[f(t)]$ as $K[f(t)] = \Sigma_{\{i=1\}}^{\infty} B_i v^i$, then identifying powers of v gives the relation $B_i = C_{\{i-1\}} \cdot (i-1)!$, for all $i \geq 1$. Substituting the expression for $C_{\{i-1\}}$, we obtain the explicit symbolic formula

$$B_i = (i-1)! \cdot a\,\Sigma_{\{k=0\}}^{\{i-1\}} P_{0k}\left[\frac{r^{\{i-1-k\}}}{(i-1-k)!} + \delta_{\{i-1-k,odd\}} \cdot \frac{r^{\{i-1-k\}}}{(i-1-k)!}\right], \; i \geq 1.$$

Equivalently,

$$B_i = a\,\Sigma_{\{k=0\}}^{\{i-1\}} P_{ok}\left[\frac{r^{\{i-1-k\}(i-1)!}}{(i-1-k)!} + \delta_{\{i-1-k,odd\}} \cdot \frac{r^{\{i-1-k\}(i-1)!}}{(i-1-k)!}\right].$$

9.4 Symbolic form of encryption rule

Given a fixed integer $p \in \mathbb{N}$ (for example $p = 3$), the first-iteration cipher coefficients P_{1i} and keys L_{1i} are defined symbolically as: $P_{1i} = (B_{\mathrm{i}} + p) mod\, 26$, for $i = 0,1,2,\dots, L_{1i} = \frac{B_{\mathrm{i}} + p - P_{1j}}{26}$. In practice, the sum for B_{i} is finite because $P_{0k} = 0$ for $k > n$, and one usually truncates at some maximal index $i = d$ associated with the effective degree of the transform polynomial used to encode the ciphertext.

9.5 General $k-th$ iteration for arbitrary a and r

For higher iterations, we can repeat the construction. Assume that after $(k-1)$ iterations we have a sequence $P_{\{k-1,i\}}$ with finite support, and define the generating function $f_{\{k-1\}(t)} = a\, P_{\{k-1\}(e^{\{rt\}} + \sinh(rt))} = \Sigma_{\mathrm{m}}\, C^{\{(k-1)\}}{}_{\mathrm{m}}\, t^{\mathrm{m}}$. Exactly as before we obtain $C^{\{(k-1)\}}{}_{\mathrm{m}} = a\, \Sigma_{\{j=0\}}^{m} P_{\{k-1,j\}E_{\{m-j\}}}$, and the Kamal-transform coefficients $B^{\{(k)\}_{\mathrm{i}}} = C_{\{i-1\}}^{\{(k-1)\}} \cdot (i-1)!\,,\ i \geq 1$. The $k-th$ iteration ciphertext is then given by $P_{\{k,i\}} = \left(B^{\{(k)\}_{\mathrm{i}}} + p\right) mod\, 26,\ \ L^{\{(k)\}_{\mathrm{i}}} = \frac{B^{\{(k)\}_{\mathrm{i}}} + p - P_{\{k,i\}}}{26}$.

Theorem 3.1 (General first-iteration Kamal–transform cipher)

Let the plaintext be encoded as a finite sequence $P_{0i}, i = 0, \dots, n$, with $P_{0i} = 0$ for $i > n$, and let $a, r, p \in \mathbb{N}$. Define C_{m} and B_{i} as above, with

$$C_{\mathrm{m}} = a\, \Sigma_{\{k=0\}}^{m} P^{0}{}_{\mathrm{k}}\, E_{\{m-k\}},$$

$$E_{\{m-k\}} = \frac{r^{\{m-k\}}}{(m-k)!} + \delta_{\{m-k,odd\}} \cdot \frac{r^{\{m-k\}}}{(m-k)!},$$

$$B_{\mathrm{i}} = C_{\{i-1\}(i-1)!},\ \ i \geq 1 .$$

Then the first-iteration ciphertext sequence P_{1i} and key sequence L_{1i} are given by $P_{1i} = (B_{\mathrm{i}} + p)\ mod\ 26,\ \ L_{1i} = \frac{B_{\mathrm{i}} + p - P_{i1}}{26}$.

Theorem 3.2 (General $k-th$ iteration Kamal–transform cipher)

Under the same assumptions, suppose that for iteration $(k-1)$ we have $P_{\{k-1,i\}}$ with finite support. Define $C^{\{(k-1)\}}{}_{\mathrm{m}} = a\, \Sigma_{\{j=0\}}^{m} P_{\{k-1,j\}E_{\{m-j\}}}$ and $B^{\{(k)\}_{\mathrm{i}}} = C_{\{i-1\}(i-1)!}^{\{(k-1)\}}$. Then the $k-th$ iteration ciphertext and key are $P_{\{k,i\}} = \left(B^{\{(k)\}_{\mathrm{i}}} + p\right) mod\, 26,\ \ L^{\{(k)\}_{\mathrm{i}}} = \frac{B^{\{(k)\}_{\mathrm{i}}} + p - P_{\{k,i\}}}{26}$.

9.6 Decryption in the general symbolic setting

From the definition of $L^{1}{}_{\mathrm{i}}$ we have $B_{\mathrm{i}} = 26L^{1}{}_{\mathrm{i}} + P^{1}{}_{\mathrm{i}} - p$. Therefore, given $(P^{1}{}_{\mathrm{i}}, L^{1}{}_{\mathrm{i}}, p)$ one can first reconstruct all B_{i}. Since $B_{\mathrm{i}} = C_{\{i-1\}(i-1)!}$, one obtains $C_{\{i-1\}}$ by dividing by $(i-1)!$.

The resulting sequence $\{C^0, C^1, \ldots\}$ determines the transformed polynomial coefficients, which satisfy $C_m = a\,\Sigma_{\{k=0\}}^{m} P_{0k} E_{\{m-k\}},\ m \geq 0$. This is a triangular linear system in the unknowns $P^{00}, P^{01}, \ldots, P^{0}{}_{n}$, with coefficients depending on a and r through $E_{\{m-k\}}$. Since $E^0 = 1$ and $a \neq 0$, the system can be inverted recursively, starting from $m = 0$:

$$m = 0:\ C^0 = a\,P^{00}E^0 = a\,P^{00} \Rightarrow P^{00} = \frac{C^0}{a}.$$

$$m = 1:\ C^1 = a(P^{00}E^1 + P^{01}E^0) \Rightarrow P^{01} = \frac{\frac{C^1}{a} - P^{00}E^1}{E^0}.$$

$$m = 2:\ C^2 = a(P^{00}E^2 + P^{01}E^1 + P^{02}E^0) \Rightarrow P^{02} = \frac{\frac{C^2}{a} - P^{00}E^2 - P^{01}E^1}{E^0}.$$

Continuing in this way, one recovers all plaintext coefficients P_{0i}. The same procedure applies to higher iterations: given $\left(P_{\{k,i\}}, L^{\{(k)\}_i}\right)$, one reconstructs $P_{\{k-1,i\}}$ and eventually the original plaintext P_{0i}.

Part 3 provides fully general symbolic formulas for the Kamal-transform-based cryptosystem with arbitrary parameters a and r, expressed in closed symbolic form and ready to be particularized to concrete numerical examples.

PART 4: DETAILED NUMERICAL EXAMPLES AND CODE IMPLEMENTATION

10. Numerical Examples Under Kamal Transform

10.1 Example 1: Plaintext NETWORK, $a = 1, r = 2, p = 3$

Step 1: Encode the plaintext NETWORK using the standard mapping $a = 0, b = 1, \ldots, z = 25$

N = 13, E = 4, T = 19, W = 22, O = 14, R = 17, K = 10.

Thus the initial sequence (iteration $k = 0$) is $P^{00} = 13, P^{01} = 4, P^{02} = 19, P^{03} = 22, P^{04} = 14, P^{05} = 17, P^{06} = 10$, and $P^{0}{}_{i} = 0$ for all $i \geq 7$.

Step 2: Form the generating function $f_{0(t)} = P^0\left(e^{\{2t\}} + \sinh(2t)\right)$.

Expanding $e^{\{2t\}}$ and $\sinh(2t)$ in Taylor series gives $e^{\{2t\}} = \frac{\Sigma_{\{n=0\}}^{\infty}(2t)^n}{n!}$, $\sinh(2t) = \frac{\Sigma_{\{n=0\}}^{\infty}(2t)^{\{2n+1\}}}{(2n+1)!}$.

Multiplying these series by the polynomial $P^0(t) = \Sigma_i P^{0}{}_{i}\, t_i$ and collecting like powers of t yields a polynomial of finite degree (because only finitely many $P^{0}{}_{i}$ are non-zero). Following the same algebraic steps as in the original paper leads to the coefficients of the transform-domain polynomial.

Step 3: Apply the Kamal Transform and use the property $K[t^n] = n!\ v^{\{n+1\}}$. Dividing by v and reading off the coefficients of the resulting polynomial in v reproduces the following values. For NETWORK one obtains the following coefficients B_i up to degree 13:

i	B_i	$B_i + p\ (p = 3)$	$P^1{}_i = (B_i + p)\ mod\ 26$	$L^1{}_i = \frac{B_i + p - P_{1i}}{26}$
0	13	16	16	0
1	34	37	11	1
2	76	79	1	3
3	208	211	3	8
4	224	227	19	8
5	1152	1155	11	44
6	640	643	19	24
7	2816	2819	11	108
8	0	3	3	0
9	7168	7171	21	275
10	0	3	3	0
11	34816	34819	5	1339
12	0	3	3	0
13	81920	81923	23	3150

The sequence $\{P^1{}_i\}$ corresponding to the first iteration is therefore $P^{10} = 16, P^{11} = 11, P^{12} = 1, P^{13} = 3, P^{14} = 19, P^{15} = 11, P^{16} = 19, P^{17} = 11, P^{18} = 3, P^{19} = 21, P^{110} = 3, P^{111} = 5, P^{112} = 3, P^{113} = 23.$

Mapping back to letters using A = 0, B = 1, …, Z = 25 gives the ciphertext QLBDTLTLDVDFDX.

Thus we write concisely: NETWORK → QLBDTLTLDVDFDX (under Kamal-transform-based scheme).

Note that the length of the ciphertext is degree $\left(f^0(t)\right) + 1$, exactly as in the original model. The difference lies only in the interpretation of the underlying integral transform that produces the polynomial coefficients.

10.2 Further Examples (Same Structure, Different Parameters)

As in the original work, we now list additional numerical examples for different choices of parameters a and r, while still using the Kamal Transform as the underlying operator.

1) For $a = 1, r = 2, p = 3$, plaintext NETWORK becomes ciphertext: QLBDTLTLDVDFDX.
2) For $a = 1, r = 1, p = 3$, plaintext NETWORK becomes ciphertext: QUWDRNNZDRDUDN.
3) For $a = 3, r = 3, p = 3$, plaintext ENCRYPTION becomes ciphertext: PAFPLUIURKDSDSDBDRDQ.
4) For $a = 2, r = 2, p = 3$, plaintext ENCRYPTION becomes ciphertext: LTTPRZRHVJDFDBDBDPDD.
5) For $a = 1, r = 1, p = 3$, plaintext TRANSFORMATION becomes ciphertext: WNDHVIRHPVWQREDUDPDDDWDLDRDQ.
6) For $a = 2, r = 2, p = 3$, plaintext TRANSFORMATION becomes ciphertext: PRDPHLBNLBTDFHDPDRDDDLDLDHDD.

Each of these mappings is obtained by repeating the same steps: encode the plaintext, construct $f(t) = aP\left(e^{\{rt\}} + \sinh(rt)\right)$, apply Kamal Transform, read off polynomial coefficients, apply the modular rule $P_{1\mathrm{i}} = (B_\mathrm{i} + p)\ mod\ 26$, and finally decode.

Acknowledgment

The authors gratefully acknowledge the foundational structure of the original Sumudu-transform-based paper and the introduction of the Kamal Transform in the second uploaded reference, which together motivated the present Kamal-transform-based cryptographic framework.

References:

1) Lokenath Debnath and D. Bhatta. Integral transform and their application second Edition, Chapman & Hall /CRC (2006).
2) G. K. Watugala, Sumudu transform- A new integral transform to Solve differential equation and control engineering problems. Math. Engrg Induct .6 (1998), no 4,319-329.
3) A. Kilicman and H. E. Gadain. An application of double Laplace transform and sumudu transform, Lobachevskii J. Math.30 (3) (2009), pp.214-223.

4) J. Zhang, A Sumudu based algorithm m for solving differential equations, Comp. Sci. Moldova 15(3) (2007), pp – 303-313.
5) A. Kashuri and A. Fundo, "A New Integral Transform," Advances in Theoretical and Applied Mathematics, Vol. 8, No. 1, 2013, pp. 27-43.
6) Tarig M. Elzaki, (2011), The New Integral Transform "Elzaki Transform" Global Journal of Pure and Applied Mathematics, ISSN 0973-1768, Number 1, pp. 57-64.
7) K. S. Aboodh, The New Integral Transform "Aboodh Transform" Global Journal of pure and Applied Mathematics, 9(1), 35-43(2013).
8) Dr. B. S Grewal, Higher Engineering Mathematics- 42nd edition (Khanna Publishers).
9) Hassan Eltayeb and Adem K. I., A Note on the Sumudu Transforms and Differential Equations, Applied Mathematical Sciences, 2010, 4(22),1089-1098.
10) Dhanorkar, G. A. and Hiwarekar, A. P., (2011), A Generalized Hill Cipher Using Matrix Transformation, International Journal of Mathematical Science and Engineering Applications., 5, pp. 19-23.
11) Douglas, R. S., (2011), Cryptography Theory and Practice, 3rd Edition, Chapman and Hall/CRC.
12) Naga Lakshmi, G., Ravi Kumar, B. and Chandra Shekhar, A., (2011), A cryptographic Scheme of Laplace Transforms, International Journal of Mathematical Archieve-2, pp. 2515-2519.
13) Singh S. The code book: The science of secrecy from ancient Egypt to quantum cryptography. Doubleday: New York, United States, 1999.
14) Shannon CE. Communication theory of secrecy systems. The Bell System Technical Journal, 1949, 28(4), 656-715.
15) Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 1978, 21(2), 120-1266.
16) Daemen J, Rijmen V. The design of rijndael: AES - The advanced encryption standard, 1st ed. Springer Berlin: Heidelberg, Germany, 2002.
17) Oppenheim A V, Schafer RW. Discrete-Time signal processing, 2nd ed. Prentice Hall: New Jersey, United States, 1999.

18) Sasirekha N, Hemalatha M. An enhanced code encryption approach with HNT transformations for software security. International Journal of Computer Applications, 2012, 53(10).

19) Donald E. K. The art of computer programming, Volume 2: Semi-numerical Algorithms, 3rd ed. Addison-Wesley: Boston, United States, 1997.

20) Koblitz N. Elliptic curve cryptosystems. Mathematics of Computation, 1987, 48(177), 203-209. DOI: 10.1090/S0025-5718-1987-0866109-5.