

Supplementary Materials

An Operational DNA Strand Displacement Encryption Approach

Enqiang Zhu ¹, Xianhang Luo ¹, Chanjuan Liu ^{2,*} and Congzhou Chen ³

¹ Institute of Computing Science and Technology, Guangzhou University, Guangzhou 510006, China; zhuenqiang@gzhu.edu.cn (E.Z.); 2112006164@e.gzhu.edu.cn (X.L.)

² School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China

³ School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China; chencongzhou@pku.edu.cn

* Correspondence: chanjuanliu@dlut.edu.cn

Supplementary Materials:

Table S1. DNA coding.

Group 1 Code-G			Group 2 Code-TT			Group 3 Code-TA		
No.	Character	DNA codon	Character	DNA codon	Character	DNA codon		
1	Space	AT	n	AT	.	AT		
2	e	CT	s	CT	u	CT		
3	shift	TC	h	TC	,	TC		
4	t	TG	r	TG	w	TG		
5	a	AC	d	AC	m	AC		
6	o	AG	l	AG	f	AG		
7	i	CG	c	CG	y	CG		
8	g	AAT	3	AAT	;	AAT		
9	p	AAC	4	AAC	q	AAC		
10	b	AAG	5	AAG	z	AAG		
11	v	CAT	6	CAT	<	CAT		
12	-	CAA	7	CAA	=	CAA		
13	(CAC	8	CAC	%	CAC		
14)	CAG	9	CAG	+	CAG		
15	k	CCA	j	CCA	*	CCA		
16	0	CCT	x	CCT	?	CCT		
17	1	CCG	/	CCG	>	CCG		
18	2	CCC	:	CCC	tab	CCC		
19	return	AAAT	\$	AAAT	{	AAAT		
20	^	AAAA	&	AAAA	}	AAAA		
21	-	AAAC	~	AAAC	"	AAAC		
22	#	AAAGC	[AAAGC	\	AAAGC		
23	@	AAAGT]	AAAGT		AAAGT		
24	!	GTCGCCG						
25	Page break	GTCTACCC						

Table S2. DNA encoding and decoding rules.

Rule	1	2	3	4	5	6	7	8
00	A	A	T	T	G	G	C	C
01	G	C	G	C	A	T	A	T
10	C	G	C	G	T	A	T	A
11	T	T	A	A	C	C	G	G

Table S3. Synthetic DNA complexes.

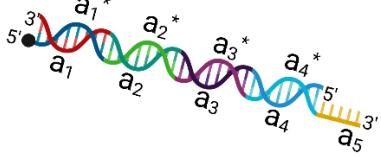
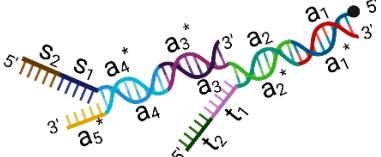
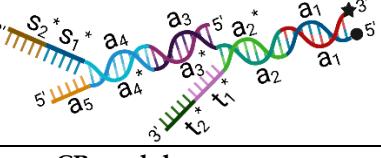
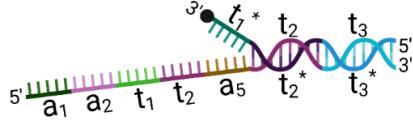
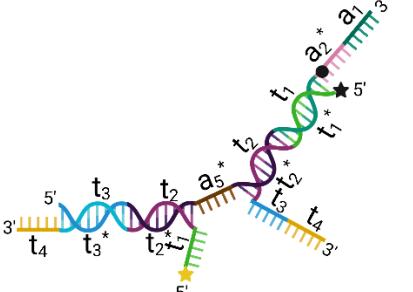
		DR-module
	B	
	D	
	G	
		CR-module
	B	
	D	

Table S4. DNA sequence design.

DNA sequence design of DR-module			
NO.	Name	Sequences(5'-3')	
B-d/ D-d	a ₁ *a ₂ *a ₃ *a ₄ *	GAGTAATTGTGTGGAGATGTGGAGTATTAGGAGTTGAAGGATTGAG-TGGTGGAGTA	
B-u	a ₁ a ₂ a ₃ a ₄	CACCACTCAATCCTCCAAACTCCTAATACTCCACTCTAC-CACATCTCCACACACAAATTACTC	
A	a ₁ a ₂ a ₃ a ₄ a ₅	TACTC CACCACTCAATCCTCCAAACTCCTAATACTCCACTCTACCACATCTCACACACAAATTACT C	
D-u1	s ₂ s ₁ a ₄ a ₄	TTAGGAGATC CACCACTCAATCCTCCAAACTCCTAATACTC	
D-u2	t ₂ t ₁ a ₂ a ₁	AGGGTGGITA CACTCTACCACATCTC CACACACAAATTACTC	
G-u1	a ₃ *a ₄ *s ₁ *s ₂ *	GAGTATTAGGAGTTGGAGGATTGAGTGG CATCT CCTAA	
G-u2	a ₁ *a ₂ *t ₁ *t ₂ *	GAGTAATTGTGTGGAGATGTGGTAGAGTGTAACCACCCT	
DNA sequence design of CR-module			
NO.	Name	Sequences(5'-3')	
B-d	a ₁ a ₂ t ₁ t ₂ a ₅ t ₂ t ₃	CCTAACATCTTACTC CACTCTACCACATCTCCAAACTCCTAATACTCCACTCTAC-CACATCTCCACACACAAATTACTC	
B-u	t ₁ *t ₂ *t ₃ *	GAGTAATTGTGTGGAGATGTGGAGT GAGTA	
A/ D-u1/ D-u2	t ₁ t ₂ t ₃ t ₄	TACTC CACTCTACCACATCTCCACACACAAATTACTCCACCACACTCAATCCTTC	
D-d	a ₁ *a ₂ *t ₁ *t ₂ *a ₅ *t ₂ *	GAGTAATTGTGT GTGGAGATGTGGTAGAGTGGAGTATTAGGAGTTGGAGATGTGG-TAGAGTG GAGTA AGATG TTAGG	

Table S5. DNA XOR operation.

XOR	A	G	C	T
A	A	G	C	T
G	G	A	T	C
C	C	C	A	G
T	T	T	G	A

Table S6. DNA ADD operation.

ADD	A	G	C	T
A	A	G	C	T
G	G	A	T	C
C	C	T	G	A
T	T	C	A	G

Table S7. Example of groupCS.

To illustrate the method, consider the following example, where the example 1 illustrates the case that the length of D_1 meet the requirements after the first k -round shift, while the example 2 illustrates the case that the length of D_1 does not meet the requirements after the first k -round shift.

Example 1

1. Let $D_0=AGTTACCGCG$, i.e., $l_0=10$, and we would like to extend D_0 to a new DNA sequence D_1 with length l' , $l' \geq l = 20$.

Transform D_0 into (0,1)-sequence S according to Table S2, $S=00011110110110001101$. Then, we set $k = 2\lceil l/l_0 \rceil = 4$ and perform k rounds of cyclic shift; see the following.

Initial	$S = 0001111001010011001$
Round 1	$S = 0011111001010011001$ $Q = 0011111001010011001$
Round 2	$S = 11111001010011001000$ $Q = 001111100101001100101111001010011001000$
Round 3	$S = 11110010100110010001$ $Q = 00111110010100110010111100101001100100011110010100110010001$
Round 4	$S = 11001010011001000111$ $Q = 00111110010100110010111100101001100100011110010100110010001111$

Now, divide Q into $\lceil \frac{80}{8} \rceil = 10$ groups, denoted by $Q_1=00111110$, $Q_2=01010011$, $Q_3=00101111$, $Q_4=10010100$,

3. $Q_5=11001000$, $Q_6=11110010$, $Q_7=10001101$, $Q_8=00011100$, $Q_9=10100110$, $Q_{10}=01000111$. Then, delete Q_2, Q_3, Q_6, Q_7 and Q_{10} , and transform the other groups into DNA sequence $D_1=ATTGTAAGGCTCAGTAAATA$.

4. The requirements $l' \geq l$ is satisfied, and output D_1 .

Example 2

1. Let $D_0=AGTCTGCATG$, i.e., $l_0=10$, and we would like to extend D_0 to a new DNA sequence D_1 with length l' , $l' \geq l = 20$.

Transform D_0 into (0,1)-sequence S according to Table S2, $S=00011110110110001101$. Then, we set $k = 2\lceil l/l_0 \rceil = 4$ and perform k rounds of cyclic shift; see the following.

Initial	$S = 00011110110110001101$
Round 1	$S = 00111101101100011010$ $Q = 00111101101100011010$
Round 2	$S = 11110110110001101000$ $Q = 00111101101100011010110110110001101000$
Round 3	$S = 11101101100011010001$ $Q = 0011110110110001101011011011000110100011010001$
Round 4	$S = 10110110001101000111$ $Q = 00111101101100011010111011011000110100011010001111$

Now, divide Q into $\lceil \frac{80}{8} \rceil = 10$ groups, denoted by $Q_1=00111101$, $Q_2=10110001$, $Q_3=10101111$, $Q_4=01101100$, $Q_5=01101000$,

3. $Q_6=11101101$, $Q_7=10001101$, $Q_8=00011011$, $Q_9=01100011$, $Q_{10}=01000111$. Then, delete $Q_1, Q_3, Q_6, Q_7, Q_8, Q_9$ and Q_{10} , and transform the other groups into DNA sequence $D_1=ACGTCGATCGGT$.

The length of D_1 is 12, $l'=12 < l$, which does not meet the requirements. So, add k rounds of cyclic shift. $k = 2\lceil \frac{l-l'}{l_0} \rceil = 2$.

See the following.

Initial	$S = 10110110001101000111$
Round 1	$S = 01101100011010001111$ $Q = 01101100011010001111$
Round 2	$S = 10110001101000111101$ $Q = 0110110001101000111101$

Now, divide Q into $\lceil \frac{40}{8} \rceil = 5$ groups, denoted by $Q_1=01101100$, $Q_2=01101000$, $Q_3=11111011$, $Q_4=00011010$, $Q_5=00111101$.

5. Then, delete Q_3, Q_4 and Q_5 , and transform the other groups into DNA sequence and splice them after D_1 . D_1 was extended to $ACGTCGATCGGT$. Now the length of D_1 is 20, $l'=20 = l$.

6. The requirements $l' \geq l$ is satisfied, and output D_1 .

Table S8. An example of the algorithm BioEN.

1.	Encrypted string abc. Transform abc into a DNA sequence D ₁ according to the tri-phase transformation. D ₁ =GCG-GATTAA.
2.	Set 2-11-2 as the seed for the initial key. Transform it into DNA sequence according to the rules listed in Supplementary Table S1, and then extend it to a new DNA sequence D ₂ with length at least that of D ₁ according to groupCS. D ₂ =GAACGCCGCC.
3.	Divide D ₂ into three groups, where group2-1=GAAC, group2-2=GCCC, group2-3=GCCC. According to the middle two digits of the 01 sequence of 8 bits corresponding to each group, label group 1 by <i>add</i> , group 2 by <i>xor</i> , and group 3 by <i>xor</i> .
4.	Divide D ₁ into three groups, where group1-1=GCGG, group1-2=ATTT, group1-3=TAA.
5.	Conduct <i>add</i> operation between group1-1 and group2-1; Conduct <i>xor</i> operation between group1-2 and group2-2; and conduct <i>xor</i> operation between group1-3 and group2-3. The result D ₃ =ACGTGGGTCC.
6.	Then, we obtain the ciphertext C= U è STX by transforming D ₃ into ASC II code by Supplementary table S2.

Table S9. Proof of Key Space Analysis.

1.	Obtain the key 2-11-2 through experiments. Transform the key into DNA sequence X according to the rules listed in Table S1. X=GCCCCAAGCCGGCGCAAGCCC. Transform X into (0,1)-sequence S according to the rules in the first column of Table S2. S=01101010011000001101001011010010110000001101010.
1.	Use R ⁱ (S) to represent the generated (0,1)-sequence after the i th round of shift. i starts from 1, and i+1 after shift one round. R ⁰ (S) = S.
	If i ≡ 1 (mod 2), then R ⁱ (S) moves the first place to the last place on the basis of R ⁱ⁻¹ (S); if i ≡ 0 (mod 2), then R ⁱ (S) moves the first two digits to the last two digits on the basis of R ⁱ⁻¹ (S).
	R ¹ (S) 110101001100000011010010110100101100000011010100
	R ² (S) 010100110000001101001011010010110000001101010011
2.	R ³ (S) 101001100000011010010110100101100000011010100110
	...
	R ³² (S) 011010100110000001101001011010010110000001101010
	R ³³ (S) 110101001100000011010010110100101100000011010100
	R ³⁴ (S) 010100110000001101001011010010110000001101010011
	...
3.	After 32 shifts, the generated 01 sequence is repeated, i.e., R ^{32+j} (S) = R ^j (S), where j is a nonnegative positive integer. The length of the (0,1)-sequence generated by 32 rounds of shift determines the size of our key space. Each round of shift can produce a 48-bit (0,1)-sequence, and 32 rounds can produce 1536 bit (0,1)-sequence. Therefore, key space is 2 ¹⁵³⁶ .

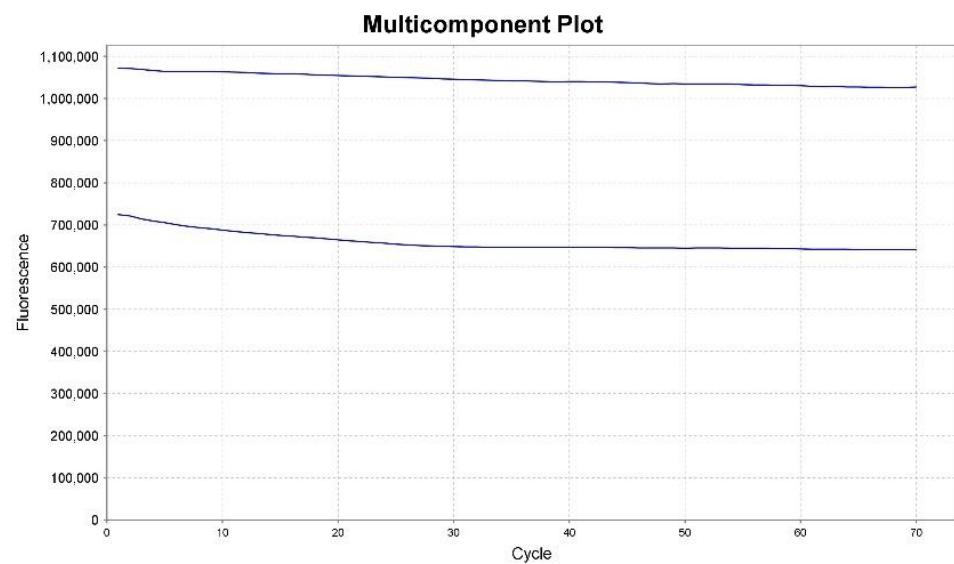


Figure S1. The fluorescence intensity changes when the concentration ratio of ABDG is 1:1:1:1.

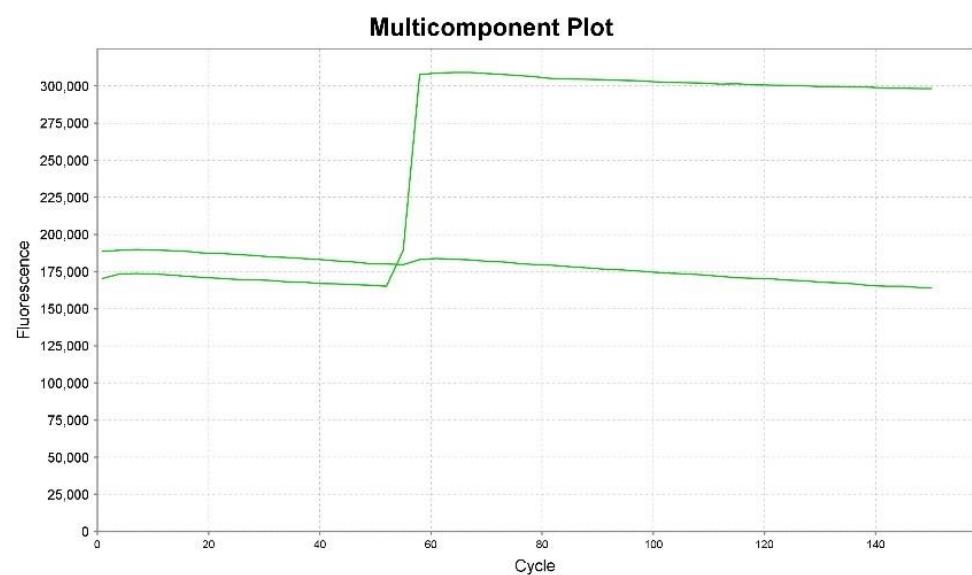


Figure S2. The fluorescence intensity changes when the concentration ratio of ABD is 1:1:1.