Visualization of Advanced Encryption Standard Cipher using CrypTool

Ivan Savka[1], Yurii Yanovskyi[2],
Ihor Lazarovych[2], Mykola Kozlenko[2]

1. Department of Analysis, Geometry and Topology, Institute for Applied Problems in Mechanics and Mathematics, UKRAINE, Lviv, 3b Naukova Street,
E-mail: ivan.savka@pnu.edu.ua

2. Department of Information Technology, Vasyl Stefanyk Precarpathian National University, UKRAINE, Ivano-Frankivsk, 57 Shevchenko Street,
E-mail: mykola.kozlenko@pnu.edu.ua

***Abstract* – This paper presents an overview of the cryptographic algorithm visualization possibilities of the CrypTool. The AES cipher is used as an example. Visualization tools for modern cryptographic algorithms in CrypTool make it possible to track the content of cryptographic transforms at every step. This makes it easier to understand the complex algorithms in software development.**

***Анотація* – Ця робота представляє огляд можливостей візуалізації криптоалгоритмів за допомогою CrypTool на прикладі шифру AES. Засоби візуалізації сучасних криптоалгоритмів у CrypTool дають можливість відслідковувати зміст криптографічних перетворень на кожному кроці. Це дозволяє полегшити розуміння "внутрішньої" суті складних алгоритмів при розробці програмного забезпечення.**

Keywords – CrypTool, AES, Symmetric Block Cipher, Cryptographic Algorithm.

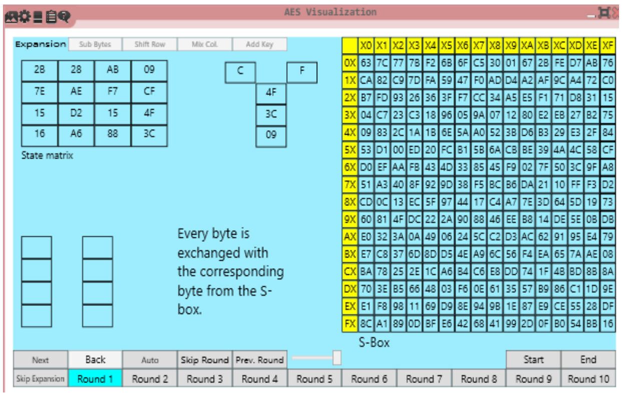
I. Introduction

CrypTool is free and open source educational software that illustrates cryptographic and crypto analytic concepts. It enables better understanding of encryption / decryption algorithms. It teaches users how to prevent network threats and ensure the security of their data [1]. The project started by Professor Bernhard Esslinger in 1998 to improve the skills of the bank's employees in cyber security and cryptography. It was developed by several German universities (Darmstadt, Duisburg-Essen, Siegen). There are five versions of the CrypTool: CT1, CT2, JCT, CTO, and MTC3, which are used in different aspects [1]. In particular, in the article [2] one can find a detailed overview and comparative analysis of versions. The program contains classical and modern encryption techniques, including symmetric, asymmetric and hydride cryptographic algorithms, hashing functions, digital signatures and other features. It can demonstrate the threats and risks that may arise from the use of cryptographic protection, the use of crypto analysis, and known attacks on cryptographic systems. The implemented mathematical functionality allows to determine whether a number is prime, to generate prime numbers in a given range, to decompose a number into prime factors (factorization of a number), to calculate entropy and autocorrelation, to calculate the frequency of a symbol or sequence of symbols in a text. It is possible to develop custom plug-ins in CrypTool 2 (CT2) those implement needed cryptographic algorithms or other custom functionality [3].

II. Methodology

There is a brief explanation of the principle of operation for each already implemented algorithm in the application. The user can encrypt the entered text with the appropriate parameters of the algorithm. The encrypted text will be displayed after the execution of the current workspace of the corresponding algorithm. Let us consider the capabilities of the CrypTool 2 toolkit as an example of an Advanced Encryption Standard (AES) cipher. AES is a modern standard for symmetric block ciphers and is actively used in protocols such as SSL, SSH, Wireless LAN 802.11i, etc. When describing the algorithm, the Galois field GF(2^8) is used, constructed as an extension of the field GF(2) by the roots of some irreducible polynomial [4].

One can use the appropriate template to open the template in the CT2 version. It is possible to find it, for example, by entering the "AES" in the template filter. In particular, the AES visualization template reproduces the step-by-step process of encrypting a 128-bit message [5].



2B	28	AB	09
7E	AE	F7	CF
15	D2	15	4F
16	A6	88	3C

State matrix

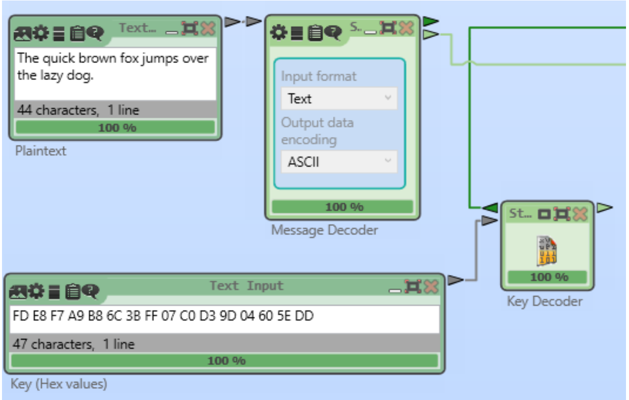
Every byte is exchanged with the corresponding byte from the S-box.

	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	XA	XB	XC	XD	XE	XF
0X	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1X	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2X	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3X	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4X	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5X	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6X	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7X	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8X	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9X	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
AX	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
BX	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
CX	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
DX	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
EX	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
FX	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

S-Box

Fig. 1. AES Visualization

Another AES Cipher template (input text) can be used to encrypt arbitrary text.



The quick brown fox jumps over the lazy dog.

44 characters, 1 line

100 %

Input format: Text

Output data encoding: ASCII

100 %

100 %

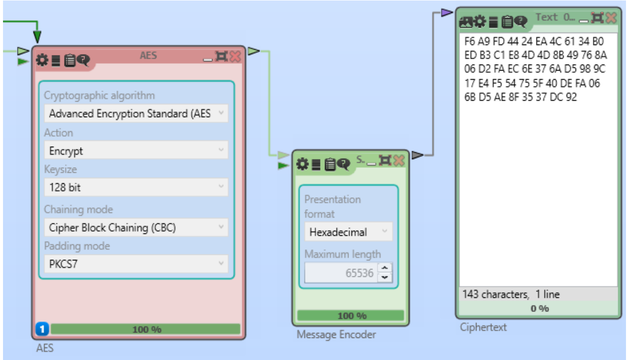
FD E8 F7 A9 B8 6C 3B FF 07 C0 D3 9D 04 60 5E DD

47 characters, 1 line

100 %

Fig. 2. AES Cipher Template (input text). Part 1.

Plain text is entered in the Plaintext component, and the corresponding encrypted text is obtained in the Ciphertext component after starting the template. The AES component works with bytes, so the source text is first converted into bytes using the Message Decoder component. It is possible to configure the plaintext format and encoding in this component. The resulting encrypted text sequence is converted to hexadecimal format with the Message Encoder component. The template can be used for message decrypting as well.



AES

Cryptographic algorithm: Advanced Encryption Standard (AES

Action: Encrypt

Keysize: 128 bit

Chaining mode: Cipher Block Chaining (CBC)

Padding mode: PKCS7

Presentation format: Hexadecimal

Maximum length: 65536

```
F6 A9 FD 44 24 EA 4C 61 34 B0
ED B3 C1 E8 4D 4D 8B 49 76 8A
06 D2 FA EC 6E 37 6A D5 98 9C
17 E4 F5 54 75 5F 40 DE FA 06
6B D5 AE 8F 35 37 DC 92
```

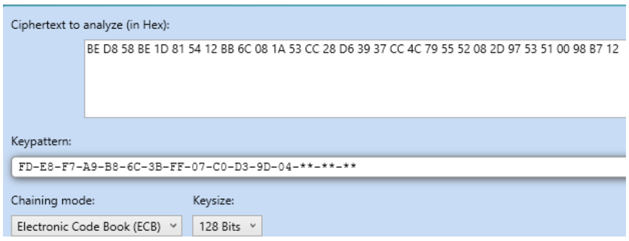
143 characters, 1 line

Fig. 3. AES Cipher Template (input text). Part 2.

The AES component has several parameters and uses a key. Such parameters are the following:

- Cryptographic algorithm: this is the AES / Rijndael mode
- Action: encrypt or decrypt
- Key size: the number of bits in the key: 128/192/256
- Chaining mode: how encrypted data from one block is used in the next
- Padding mode: for example, filling blocks with zero is smaller than the size of the AES block.

On the other hand, CrypTool contains a cryptographic analysis of the AES algorithm, in particular AES - Ciphertext-only analysis.



Ciphertext to analyze (in Hex):

```
BE D8 58 BE 1D 81 54 12 BB 6C 08 1A 53 CC 28 D6 39 37 CC 4C 79 55 52 08 2D 97 53 51 00 98 B7 12
```

Keypattern:

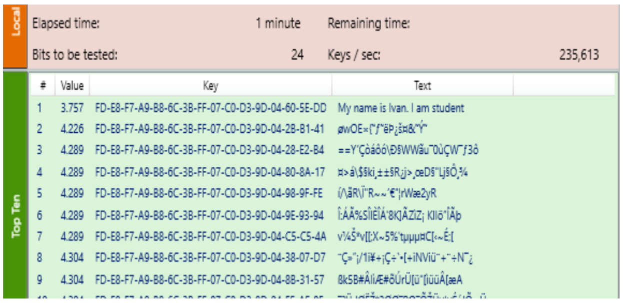
```
FD-E8-F7-A9-B8-6C-3B-FF-07-C0-D3-9D-04-**-**-**
```

Chaining mode: Electronic Code Book (ECB)

Keysize: 128 Bits

Fig. 4. AES – Ciphertext-only analysis.

For example, cryptographic decryption of some text subject to partial key information takes one minute (Intel Core i5 2.7 GHz 4 GB of RAM). In total, about 16.8 million possible keys are sorted. The attack uses simple entropy: encrypted text is more chaotic than the text in any human language [6].



Elapsed time: 1 minute

Remaining time:

Bits to be tested: 24

Keys / sec: 235,613

#	Value	Key	Text
1	3.757	FD-E8-F7-A9-B8-6C-3B-FF-07-C0-D3-9D-04-60-5E-DD	My name is Ivan. I am student
2	4.226	FD-E8-F7-A9-B8-6C-3B-FF-07-C0-D3-9D-04-2B-B1-41	øwOE×{"ƒ"ëÞ¿š¤&"Ý"
3	4.289	FD-E8-F7-A9-B8-6C-3B-FF-07-C0-D3-9D-04-28-E2-B4	==Y'ÇòáôóÐ§WWãu¯0ùÇW¯ƒ3ô
4	4.289	FD-E8-F7-A9-B8-6C-3B-FF-07-C0-D3-9D-04-80-8A-17	¤>á\$§ki¸±±§R¡j>¸œD§"Lj§Ó¸¾
5	4.289	FD-E8-F7-A9-B8-6C-3B-FF-07-C0-D3-9D-04-98-9F-FE	í/\ãR\Ï"R~~´€"¦rWæ2yR
6	4.289	FD-E8-F7-A9-B8-6C-3B-FF-07-C0-D3-9D-04-9E-93-94	Î:ÁÄ%SÎÌÉÎÁ'8K]ÃZÍZ¡ KÌlö"ÎÄþ
7	4.289	FD-E8-F7-A9-B8-6C-3B-FF-07-C0-D3-9D-04-C5-C5-4A	v¼Š*v[[;X~5%'tµµµ¤C[‹~É;[
8	4.304	FD-E8-F7-A9-B8-6C-3B-FF-07-C0-D3-9D-04-38-07-D7	¨Ç»";/1ï¥+¡Ç÷'•[+iNViü¨+¨÷N¯¿
9	4.304	FD-E8-F7-A9-B8-6C-3B-FF-07-C0-D3-9D-04-8B-31-57	ßk5B#ÂÌìÆ#õÚrÜ[ü"[ìüüÂ[æA
10	4.304	[illegible]	[illegible]

Fig. 5. AES – Ciphertext-only analysis.

III. Conclusion

CrypTool is a convenient tool for use in cryptography. It contains both ready-to-use templates and the ability to build new custom algorithms. Built-in tools for visualization and animation of mathematical transforms and modern cryptographic algorithms give an excellent representation and facilitate the understanding of such algorithms in the information protection and development of appropriate cyber security software. Our future research will be devoted to visualization of artificial neural networks for general cyber security related problems [7], [8] and to digital signal processing [9] for physical layer security [10] related problems.

References

[1] The CrypTool Portal [Online]. Available: http://www.cryptool.org/

[2] N. Zagatska, "Review of different releases of cryptool package as tool for information resources security," *ITLT*, vol. 31, no. 5, Nov. 2012. https://doi.org/10.33407/itlt.v31i5.744

[3] S. Przybylski, A. Wacker, M. Wander, F. Enkler, P. Vacek: Plugin Developer Manual – How to build your own plugins for CrypTool 2.0. Version 0.7, July 16, 2011.

[4] Jun Ma, Jun Tao, Jean Mayo, Ching-Kuang Shene, Melissa Keranen, and Chaoli Wang. 2016. AESvisual: A Visualization Tool for the AES Cipher. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '16)*. Association for Computing Machinery, New York, NY, USA, 230–235. doi: https://doi.org/10.1145/2899415

[5] Matthias Becher, *Visualization of AES as a CrypTool 2 Plugin*, Bacherlor Thesis University of Mannheim, 2016. https://www.cryptool.org/images/ctp/documents/BA_Becher.pdf

[6] I. Savka and Yu. Yanovskyi, "Vykorystannia CrypTool na prykladi symetrychnoho shyfru AES," in *Proceedings of the 2019 Scientific Seminar on Innovative Solutions in Software Engineering*, Ivano-Frankivsk, Ukraine, Dec. 10, 2019, pp. 16-18, doi: https://doi.org/10.5281/zenodo.4084539

[7] M. Kozlenko, V. Tkachuk, and M. Dutchak, "Software implementation of microcomputer based intrusion detection and prevention system with binary neural network," in *Proceedings 2nd International Scientific-Practical Conference on Problems of Cyber Security of Information and Telecommunication Systems (PCSITS)*, O. Oksiiuk et al, Eds. Taras Shevchenko National University of Kyiv, Kyiv, Ukraine, Apr. 11-12, 2019, pp. 371-373

[8] M. Kozlenko and V. Tkachuk, "Deep learning based detection of DNS spoofing attack," in *Proceedings of the 2019 Scientific Seminar on Innovative Solutions in Software Engineering*, Ivano-Frankivsk, Ukraine, Dec. 10, 2019, pp. 10-11, doi: https://doi.org/10.5281/zenodo.4091018

[9] M. Kozlenko, I. Lazarovych, V. Tkachuk, and V. Vialkova, "Software Demodulation of Weak Radio Signals using Convolutional Neural Network," *2020 IEEE 7th International Conference on Energy Smart Systems (ESS)*, Kyiv, Ukraine, 2020, pp. 339-342, doi: 10.1109/ESS50319.2020.9160035

[10] T. M. Hoang, T. Q. Duong, H. D. Tuan, S. Lambotharan and L. Hanzo, "Physical Layer Security: Detection of Active Eavesdropping Attacks by Support Vector Machines," in *IEEE Access*, vol. 9, pp. 31595-31607, 2021, doi: 10.1109/ACCESS.2021.3059648