KMANPUB | Decision Science and Intelligent Systems | Vol. 1 No. 2 (2024): 94-110

Article history:
Received 03 February 2025
Revised 03 March 2025
Accepted 12 March 2025
Published online 20 March 2025

On Graph-Based Cryptography in Dynamical Systems

Manouchehr. Shahamat[1*]

[1] Department of Mathematics, Dezful branch, Islamic Azad University, Dezful, Iran

*** Corresponding author email address**: m.shahamat@iau.ac.ir

A r t i c l e I n f o

Article type:
Original Research

How to cite this article:
Shahamat, M. (2024). On Graph-Based Cryptography in Dynamical Systems. *Decision Science and Intelligent Systems. 1(2), 94-110.*

A B S T R A C T

For the purpose of data storage and transmission, constrained sets are required. Thus, to convert sequences from a full shift space to a sofic shift, certain constraints must be applied. One method to address this issue is the use of finite-type codes. An (X, n)-finite-type code can be used to transform sequences from a full n-shift into sequences from the shift space X. If X is a sofic space with entropy at least equal to the logarithm of n, then an (X, n)-finite-type code exists.

Keywords: *Shift space, Cryptography, Encoding, Decoding, Entropy, Graph*

Extended Abstract

In recent years, the application of graph theory and symbolic dynamics to the domain of cryptography has garnered increasing attention, especially in the context of dynamical systems. This study addresses the problem of constructing robust and flexible cryptographic schemes using tools from symbolic dynamics, specifically focusing on shift spaces, sofic systems, and finite-type coding techniques. The necessity of constrained sets for data storage and transmission is emphasized, where full shift spaces must be transformed into sofic shifts to satisfy certain entropy conditions. To facilitate this transformation, the concept of an (X, n)-finite-type code is introduced, where X is a sofic space and n represents the size of the alphabet. A central condition for the existence of such codes is that the entropy of X must be at least log(n). This requirement is grounded in the mathematical properties of shift spaces and their corresponding languages. The proposed cryptographic method utilizes finite-state graphs labeled with input and output encodings to represent constraints and transformations, ensuring data security through structural complexity and mathematical rigor. The novelty of this work lies in its ability to generalize encryption through multiple keys and graph-based encodings, transcending the limitations of traditional symmetric and asymmetric systems.

The paper begins by reviewing the historical foundations and technical classifications of cryptographic systems. Symmetric encryption, which relies on a single shared key for both encryption and decryption, is known for its computational efficiency but suffers from the vulnerability of key exchange. In contrast, asymmetric encryption introduces public and private key pairs, offering greater security in key distribution. The proposed method diverges from these conventional models by adopting a hybrid structure that incorporates multiple encryption keys, both public and private, in a dynamic switching framework. This approach begins with three keys and allows for scalable key expansion depending on the complexity and security needs of the message. Through symbolic dynamics, each data unit (e.g., 8-bit blocks) is modeled as a sequence within a full shift space and is then encoded via sofic shift constraints. These constraints are encoded into the system through labeled graphs whose edges and vertices correspond to data symbols and transition rules. Notably, the graphs used in this model are not limited to single representations, as multiple labeled graphs can provide equivalent shift space coverage, thereby enhancing the cryptographic strength via redundancy and structural variability.

Central to the methodology is the construction of labeled directed graphs (digraphs) and the utilization of their combinatorial properties. Each graph represents a specific configuration of allowable symbol sequences, constrained by adjacency matrices and transition rules. A graph is considered right-resolving (right-deterministic) if each outgoing edge from a vertex has a unique label, a property that ensures unambiguous decoding. Similarly, left-resolving and bi-resolving graphs further reinforce the determinism of encoding and decoding processes. The encoding system defined in the paper employs two such graphs in alternating fashion. The data block is divided into partitions (keys), and each sub-block is encrypted by applying encoding functions defined on alternating graphs. This process continues in a round-robin fashion, enabling dynamic shifts in the encryption path based on predefined block lengths. The graph's degree (i.e., the number of outgoing edges per vertex) and its associated label mappings form the basis of both the input and output encodings. As each bit string from the message is passed through this graph encoding system, it is transformed into a cryptographically obfuscated form that adheres to the rules of the underlying symbolic dynamic system.

A crucial innovation in this study is the use of entropy as a criterion for determining the feasibility of finite-type codes. Entropy, in the context of shift spaces, represents the capacity or complexity of a symbolic system and is calculated based on the number of admissible words of given length. For a sofic space X to admit an (X, n)-finite-type code, it must have entropy greater than or equal to log(n). This

mathematical constraint ensures that the space is rich enough to encode all the sequences from the full shift. When this condition is not met, a workaround is proposed by selecting a rational number $q < \log(n)$, identifying a positive integer k such that the adjusted entropy $\log(n^k)/k$ satisfies the constraint. Thus, a (X, n^k)-finite-type code is constructed, which encodes blocks of length k instead of individual symbols, maintaining the integrity and feasibility of encryption. This theoretical foundation ensures that the cryptographic method can be adapted to a wide range of symbolic spaces and alphabets, enhancing its versatility across different application domains.

The process of encryption and decryption is formalized through well-defined functions that act on the edges of labeled graphs. Input and output label functions are introduced, mapping symbols to graph edges and their associated output sequences. The composition of these functions results in encoding transformations that are reversible, a necessary condition for successful decryption. The decryption process (termed "decoding" or "reading back") involves inverse mappings applied in the same alternating sequence used during encryption. Importantly, the system ensures that these mappings are bijective for each segment, preserving data integrity. The introduction of partition sets, where each partition length is divisible by 8 (to match byte structure), facilitates compatibility with binary systems in digital computers. The flexibility of the method is further extended by allowing changes in partition sets (keys) and graph mappings based on the content of the data itself, adding an adaptive layer to the cryptographic process. This dynamic encoding-decoding system enhances security by making it exceedingly difficult for adversaries to predict the mapping paths or reconstruct the original message without full knowledge of the encryption scheme, partition keys, and graph functions.

The proposed cryptographic framework offers several advantages over classical methods. First, the use of multiple keys that alternate based on content and structure introduces a level of unpredictability that strengthens resistance to cryptanalysis. Second, the user has control over the number and structure of the keys, allowing for customizable security depending on the use case. Third, each encryption session can modify both public and private keys dynamically, providing a high degree of session uniqueness and resistance to replay attacks. Fourth, the model can be generalized by increasing the number of graphs, encoding functions, or even introducing stochastic selection rules where the computer autonomously chooses encoding functions based on predefined conditions or message characteristics. These extensions pave the way for a new generation of cryptographic algorithms that integrate symbolic dynamics, graph theory, and computer science into a unified framework. Additionally, increasing the size or overlap of partitions (i.e., shared data segments) provides another dimension of complexity, making brute-force attacks computationally infeasible.

In conclusion, this research presents a novel and mathematically grounded approach to cryptography based on graph theory and symbolic dynamics. By leveraging the properties of sofic systems, entropy constraints, and labeled graphs, the proposed method enables flexible and secure data encoding mechanisms that surpass traditional binary encryption in structural complexity and adaptability. The incorporation of multiple keys, dynamic switching, and reversible encoding-decoding paths ensures a high level of security while maintaining computational feasibility. The method's reliance on established mathematical theorems ensures its theoretical soundness, while its flexible implementation strategy allows for scalability and customization in real-world digital environments. This work contributes a significant step toward the development of advanced cryptographic systems rooted in dynamic systems theory and opens new directions for research at the intersection of mathematics, computer science, and cybersecurity.

علم تصمیم گیری و سیستم‌های هوشمند دوره ۱، شماره ۲ (۱۴۰۳): ۱۱۰-۹۴

تاریخچه مقاله
دریافت شده در تاریخ ۱۵ بهمن ۱۴۰۳
اصلاح شده در تاریخ ۱۳ اسفند ۱۴۰۳
پذیرفته شده در تاریخ ۲۲ اسفند ۱۴۰۳
منتشر شده در تاریخ ۳۰ اسفند ۱۴۰۳

پیرامون رمزنگاری با گراف‌ها در دستگاه‌های دینامیکی

منوچهر شهامت[1*]

۱. گروه ریاضی، واحد دزفول، دانشگاه آزاد اسلامی، دزفول، ایران

[*]ایمیل نویسنده مسئول: m.shahamat@iau.ac.ir

اطلاعات مقاله

نوع مقاله

پژوهشی

نحوه استناد به این مقاله:

شهامت، منوچهر. (۱۴۰۳). پیرامون رمزنگاری با گراف‌ها در دستگاه‌های دینامیکی. *علم تصمیم گیری و سیستم‌های هوشمند، ۱(۲)*، ۱۱۰-۹۴.

چکیده

برای ذخیره و انتقال داده‌ها نیازمند داشتن مجموعه‌های دارای محدودیت هستیم. پس برای تبدیل دنباله‌ها از یک فضای تغییرجای کامل به یک تغییرجای سافیک، محدودیت‌هایی وجود دارند. یک روش برای رفع مشکل استفاده از کدهای متناهی است. از $(X,n)-$کد متناهی می‌توان برای تبدیل دنباله‌ها از یک $n-$تغییرجای کامل به دنباله‌هایی از فضای تغییرجای X استفاده کرد. اگر X یک فضای سافیک باشد که آنتروپی آن حداقل لگاریتم n باشد، آنگاه یک $(X,n)-$کد متناهی موجود است.

کلیدواژگان: *فضای تغییرجا، رمزنگاری، کدگذاری، بازخوانی، آنتروپی، گراف*

مقدمه

رمزنگاری[1] استفاده از روش‌های ریاضی، برای برقراری امنیت اطلاعات است. دراصل، رمزنگاری دانش تغییردادن اطلاعات به کمک فرآیند رمز است. به‌صورتی که تنها شخصی که از کلید و فرآیند آگاه است می‌تواند اطلاعات اصلی را استخراج کند و شخصی که از آن‌ها اطلاع ندارد، نمی‌تواند به اطلاعات دسترسی پیدا کند.

در رمزنگاری محتویات یک متن به‌صورت حرف به حرف و در بعضی موارد رقم دوتایی[2] به رقم دوتایی تغییر داده می‌شود و هدف تغییر محتوای متن است. رمزنگاری پیشینه طولانی و درخشان دارد که به هزاران سال پیش برمی‌گردد.

در حدود ۴۰۰۰ سال پیش مصری-ها با نوشتن پیام‌هایی حیرت انگیز با یکدیگر ارتباط برقرار می‌کردند. در این متون کدهای رمزی وجود داشتند. این کد رمز تنها توسط کاتبان برای انتقال پیام از طرف آنها مورد استفاده بود. سالها بعد دانشمندان به استفاده از رمزهای بدون حرف روی آوردند که شامل تعویض الفبای پیام با دیگر الفباها همراه قوانینی مخفی بوده است. این قوانین کلیدی برای رمزگشایی پیام بودند.

رمزنگاری به دو صورت متقارن[3] و نامتقارن[4] تقسیم می‌شوند. فرآیند رمزنگاری متقارن یک روش رمزنگاری است که از یک کلید برای رمزنگاری و رمزگشایی داده‌ها استفاده می‌کند. این قدیمی‌ترین و شناخته‌شده‌ترین روش رمزنگاری است. کلید مخفی می‌تواند کلمه، شماره، یک رشته از کارکترها یا اعداد باشد. پیام طبق قوانین فرآیند رمزنگاری در کلید تغییر می‌کند. اشخاصی که از طریق رمزنگاری متقارن در حال برقراری ارتباط هستند باید کلید را مبادله کنند تا بتوانند اطلاعات را رمزنگاری و رمزگشایی کنند.

رمزنگاری متقارن کاربردهای زیادی در دنیای امروزه داشته و طیف گسترده‌ای از مزایا دارند اما یک ایراد عمده آن مشکل انتقال کلید مورد استفاده برای رمزنگاری و رمزگشایی داده‌ها است. هنگامی‌که کلیدها بر روی یک اتصال ناامن به اشتراک گذاشته می‌شوند، در معرض دستیابی هستند. اگر یک کاربر غیرمجاز به کلید متقارن دسترسی پیدا کند، امنیت هرگونه داده رمزنگاری شده با استفاده از آن کلید به خطر می‌افتد. به همین دلیل برخی از کارشناسان امنیت، تنها فرآیند نامتقارن را پیشنهاد می‌کنند.

دو محقق دانشگاه استنفورد به نام‌های مارتین هلمن[5] و ویتفیلد دیفی[6] بنیان‌گذاران روش رمزنگاری نامتقارن بودند. این دو محقق این روش را در مقاله‌ای در سال ۱۹۷۷ میلادی معرفی کردند.

رمزنگاری نامتقارن نسبت به رمزنگاری متقارن دارای یک نکته ویژه است و آن استفاده از یک جفت کلید عمومی و خصوصی است. بین این دو کلید، یک رابطه ریاضی وجود دارد که آن‌ها را به‌عنوان یک جفت مرتبط تعریف می‌کند. کلید عمومی آدرسی است که نگهدارنده نشانه‌ها است و هر کسی می‌تواند آن را مشاهده کند. کلید خصوصی برای دسترسی به آدرس و تأیید اقدامات برای آدرس استفاده می‌شود.

رمزنگاری نامتقارن را می‌توان در دستگاه‌هایی اعمال کرد که در آن بسیاری از کاربران ممکن است نیاز به رمزنگاری و رمزگشایی پیام‌ها مانند ایمیل رمزنگاری شده داشته باشند. در این موارد می‌توان از کلید عمومی برای رمزنگاری پیام استفاده کرد و از کلید خصوصی می‌توان برای رمزگشایی آن استفاده کرد.

ارزهای رمزنگاری شده به رمزنگاری نامتقارن تکیه دارند. کاربران دارای کلیدهای عمومی و خصوصی هستند. همه می‌توانند کلیدهای عمومی را ببینند اما کلیدهای خصوصی مخفی نگه داشته می‌شوند. از دیگر کاربردهای رمزنگاری نامتقارن می‌توان به موارد زیر اشاره کرد.

[1] encryption
[2] binary digit
[3] symmetric
[4] asymmetric
[5] Martin Hellman
[6] Whitfield Diffie

۱. تأیید اعتبار داده‌ها از طریق استفاده از امضاهای دیجیتالی، به‌کمک رمزنگاری نامتقارن انجام‌پذیر است.

۲. بر اساس رمزنگاری نامتقارن، امضاهای دیجیتالی می‌توانند تضمینی در مورد وضعیت اسناد الکترونیکی یا پیام الکترونیکی ارائه دهند.

۳. رمزنگاری ایمیل‌ها برای مخفی ماندن محتویات آنها.

۴. ارائه محیطی ایمن برای رأی گیری الکترونیکی.

۵. فناوری ارز دیجیتال.

۶. کاربرد در زنجیره‌های بلوکی مانند قراردادهای هوشمند ساده و پیشرفته.

بااینکه فرآیندهای رمزنگاری متقارن برای انجام رمزنگاری و رمزگشایی از یک کلید استفاده می‌کنند، در فرآیند رمزنگاری نامتقارن از یک کلید برای رمزنگاری داده‌ها و از کلید دیگری برای رمزگشایی آن استفاده می‌کند. در دستگاه‌های نامتقارن، کلید به‌کاررفته برای رمزنگاری را به‌عنوان کلید عمومی می‌شناسند و می‌توان آن را آزادانه با دیگران به اشتراک گذاشت. از سوی دیگر، کلید به‌کاررفته برای رمزگشایی کلید خصوصی است و نباید با دیگران به اشتراک گذاشت.

رمزنگاری که ما در این مقاله استفاده می‌کنیم، بیش از یک کلید دارد. پس متقارن نیست. درواقع ما از تعدادی کلید استفاده می‌کنیم. تعدادی از کلیدها در اختیار رایانه و بقیه در اختیار فرستنده و گیرنده است. نخست با سه کلید شروع خواهیم کرد. اما با اندکی تغییر می‌توان برای افزایش ایمنی، تعداد کلیدها را به هر تعداد متناهی افزایش داد.

برای رمزنگاری داده‌ها همواره از مجموعه‌ها و فضاهایی استفاده می‌شوند که دارای محدودیت‌هایی هستند. هر داده یا به‌طور خاص هر متنی را می‌توان به‌عنوان دنباله‌ای از عناصر مجموعه $\{۰,۱\}$ درنظر گرفت. پس هر داده یک عضوی ار فضای تغییرجا کامل[۷] با الفبای ۰ و ۱ است. برای رمزنگاری داده‌های این فضا، از تغییرجاهای سافیک استفاده می‌کنیم. محدودیت‌های موجود در فضای سافیک توسط کدهای متناهی بیان می‌شوند. هر کد متناهی توسط توابع دارای برچسب[۸] بیان می‌شوند.

مقالات فراوانی مانند [۱] و [۲] در زمینه رمزنگاری وجود دارند.

پیشگفتار

در این بخش، تعاریف و قضایای مورد نیاز از [۳] ، [۴] ، [۵] و [۶] انتخاب شده است.

اطلاعات اغلب توسط دنباله‌ای از نمادهای گسسته از مجموعه‌ای متناهی بیان می‌شوند. رایانه‌ها اطلاعات را با دنباله‌هایی از ۰ و ۱ ذخیره می‌کنند. در همه‌ی این مثال‌ها، مجموعه‌ای چون $\mathcal{A}$ شامل تعداد متناهی یا شمارا نماد وجود دارد که آن را الفبا می‌نامیم. دنباله‌ی متناهی از اعضای $\mathcal{A}$ را یک کلمه یا بلوک می‌نامیم. طول کلمه تعداد سمبول‌هایی است که آن کلمه دارا می‌باشد. کلمه‌ای که دارای هیچ نمادی نباشد را یک کلمه‌ی تهی می‌نامیم. مجموعه‌ی متناهی $\mathcal{A}$ را در نظر می‌گیریم. $|\mathcal{A}|$–تغییرجای کامل[۹] مجموعه‌ی دنباله‌های از دو طرف نامتناهی از اعضای $\mathcal{A}$ است که به‌صورت $\mathcal{A}^{\mathbb{Z}}$ نمایش داده می‌شوند. اگر $x \in \mathcal{A}^{\mathbb{Z}}$ ، کلمه‌ای در x از مختص i تا مختص j را به‌صورت $x_{[i,j]} = x_i x_{i+۱} \dots x_j$ نمایش می‌دهیم.

نگاشت σ را نگاشت تغییرجا روی تغییرجا کامل $\mathcal{A}^{\mathbb{Z}}$ می‌نامیم هرگاه به ازای هر $(x_n)_{n\in\mathbb{Z}}, (x_n)_{n\in\mathbb{Z}} \in \mathcal{A}^{\mathbb{Z}}$، $\sigma(x_n)_{n\in\mathbb{Z}} = (x_{n+۱})_{n\in\mathbb{Z}}$.

[7] full shift space
[8] label
[9] full shift

زیرمجموعه‌ی X از $\mathcal{A}^{\mathbb{Z}}$ را یک فضای تغییرجا گوییم هرگاه بسته و σ-پایا[10] باشد. مجموعه‌ی همه کلمات مجاز از طول n در X را با $W_n(X)$ نشان می‌دهیم و همه‌ی کلمات مجاز X را زبان X نامیم و با $W(X)$ نشان می‌دهیم. فضای تغییرجا X را تحویل‌ناپذیر[11] گوییم هرگاه $u,v \in W(X)$، آنگاه بلوک w موجود باشد که $uwv \in W(X)$.

آنتروپی[12] فضاهای تغییرجا، ظرفیت اطلاعات یا توانایی داده‌ها را تعیین می‌کند. آنتروپی فضای تغییرجا X به‌صورت

$$h(X) = \lim_{n\to\infty} \frac{1}{n} \log|W_n(X)|$$

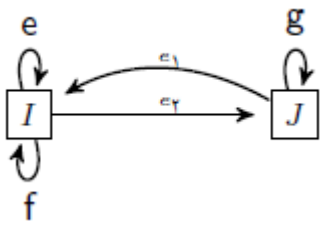
تعریف می‌شود.

گراف

گراف G شامل مجموعه‌ی $\mathcal{V} = \mathcal{V}(G)$ از رئوس و مجموعه‌ی $\mathcal{E} = \mathcal{E}(G)$ از یال‌ها را درنظر می‌گیریم. رأس ابتدایی هر یال را با $i(e) \in \mathcal{V}$ و انتهای آن را با $t(e) \in \mathcal{V}$ نشان می‌دهیم. یالی که ابتدا و انتهای آن یکسان باشد، طوقه نامیده می‌شود. هر مسیر از اتصال متوالی تعدادی یال به‌دست می‌آید. تعداد یال‌ها طول مسیر نامیده می‌شود. اگر ابتدا و انتهای مسیری یکسان باشد، آنگاه آن مسیر دور cycle نامیده می‌شود. هر طوقه دوری با طول یک است. تعداد یال‌های ورودی به یک رأس، درجه ورود و تعداد یال‌های خروجی از یک رأس، درجه خروج نامیده می‌شود. برای مثال شکل ۱ گرافی با دو رأس I و J را نشان می‌دهد که درجه ورود و خروج رأس J دو می‌باشند.

شکل ۱

مثالی برای یک گراف.



در این گراف ۳ طوقه وجود دارد.

اگر برای رأس‌های I, J، a_{IJ} برابر تعداد یال‌هایی در گراف G باشد که ابتدای آنها I و انتهای آنها J باشد، آنگاه ماتریس $A = [a_{IJ}]$ ماتریس مجاورت گراف[13] نامیده می‌شود. برای مثال ماتریس مجاورت با گراف شکل ۱ به‌صورت $\begin{bmatrix} ۱ & ۱ \\ ۱ & ۲ \end{bmatrix}$ است.

به هر ماتریس مربعی با درایه‌های نامنفی نیز می‌توان گرافی نسبت داد. چنانچه A ماتریسی مربعی از مرتبه r باشد، گراف متناظر با ماتریس A گرافی با مجموعه رأس‌های

$$\{۱,۲,\dots,r\}$$

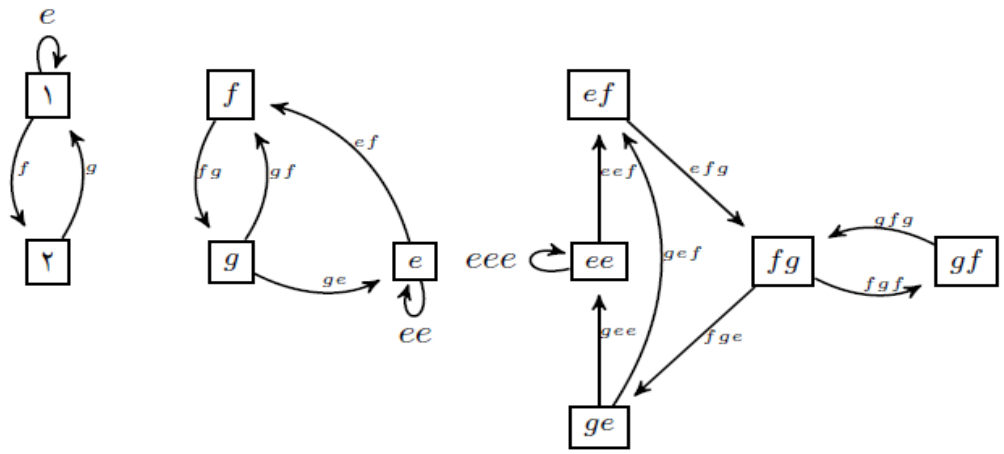
است که به تعداد a_{IJ} یال مجزا از رأس I به رأس J دارد.

[10] transitive
[11] irreducible
[12] entropy
[13] adjacencymatrix

برای هر عدد طبیعی مانند N، گراف $G^{[N]}$ که N مین گراف بالاتر[14] G نامیده می‌شود گرافی با مجموعه رأس‌های $\mathcal{V}(G^{[N]})$ است که $\mathcal{V}(G^{[N]}) = \mathcal{V}(G)$ و برای هر مسیر از طول N در گراف G از I به J، یالی از I به J وصل می‌کنیم. برای مثال در شکل ۲ گراف‌های $G^{[1]}$، $G^{[2]}$ و $G^{[3]}$ نشان داده شده است.

شکل ۲

به‌ترتیب از سمت چپ به راست گراف‌های (G^([1])، G^([2]) و G^([3]).



یک مسیر $\pi = e_1 e_2 \dots e_n$ روی گراف G دنباله‌ای از یال‌هایی است که $t(e_i) = i(e_{i+1})$. فضای تغییرجای یالی X_G فضای تغییرجایی روی $\mathcal{A} = \mathcal{E}$ است که به‌صورت زیر تعریف می‌شود.

$$X_G = \{\xi = (\xi_i)_{i\in\mathbb{Z}} \in \mathcal{E}^{\mathbb{Z}} : t(\xi_i) = i(\xi_{i+1})\}.$$

گراف G را یک پوشش فضای X_G گویند. واضح است که پوشش یک فضا منحصربه‌فرد نیست. به شکل ۳ مراجعه نمایید.

گراف برچسب‌دار[15] $\mathcal{G}$، دوتایی $(\mathcal{G}, \mathcal{L})$ است که G گرافی با مجموعه یال‌های $\mathcal{E}$ است و نگاشت $\mathcal{L}$ به هر یال e از G یک برچسب $\mathcal{L}(e)$ از $\mathcal{A}$ را نسبت می‌دهد. گراف G را گراف زیربنای $\mathcal{G}$ گوییم. برچسب مسیری مانند $\pi = e_1 e_2 \dots e_n$ در G به‌صورت

$$\mathcal{L}(\pi) = \mathcal{L}(e_1)\mathcal{L}(e_2)\dots\mathcal{L}(e_n)$$

تعریف می‌شود. اگر $\xi = \cdots e_{-1}.e_1 e_2 \dots \in X_G$، برچسب ξ را به‌صورت

$$\mathcal{L}(\xi) = \cdots \mathcal{L}(e_{-1})\mathcal{L}(e_0)\mathcal{L}(e_1)\dots$$

تعریف کرده و مجموعه‌ی همه‌ی برچسب‌های مسیرهای از دوطرف نامتناهی روی G را به‌صورت

$$X_{\mathcal{G}} := \{\mathcal{L}_\infty(\xi) : \xi \in X_G\} = \mathcal{L}_\infty(X_G).$$

نشان می‌دهیم. دراین‌حالت گراف $\mathcal{G}$ یک پوشش فضای X نامیده می‌شود.

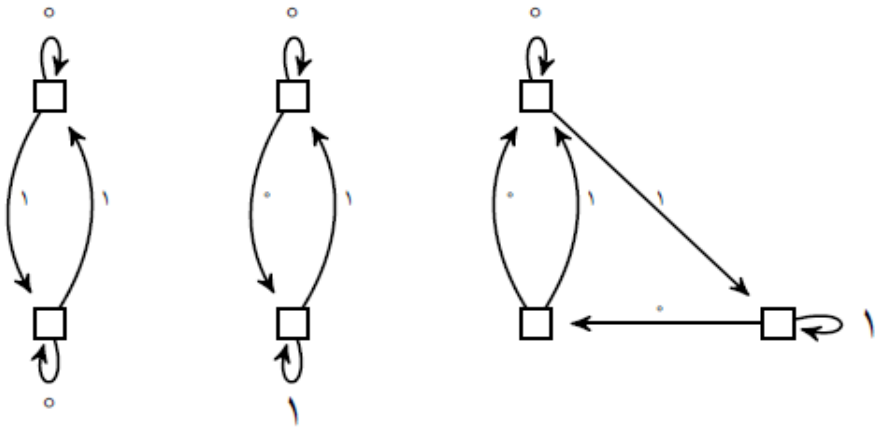
واضح است ممکن است گراف‌های متفاوتی نماینده‌ی یک فضای تغییرجا باشند. برای مثال هر سه گراف G_1، G_2 و G_3 در شکل ۳ نمایشی برای فضای تغییرجا کامل روی $\{0,1\}$ هستند. یعنی

$$X_{G_1} = X_{G_2} = X_{G_3} = \{0,1\}^{\infty}.$$

[14] Nth higher edge graph
[15] labeled graph

شکل ۳

گراف‌های متفاوت ممکن است دارای پوشش یکسان باشند. به‌ترتیب از سمت چپ به راست گراف‌های G_1، G_2 و G_3 پوشش‌هایی برای فضای تغییرجای $\infty^{\wedge}\{0,1\}$ هستند.



تعریف: *گراف G تحویل‌ناپذیر است هرگاه برای هر دو رأس I و J، مسیری از I به J وجود داشته‌باشد.*

اگر گراف پوششی فضای X تحویل‌ناپذیر باشد، آنگاه آن فضا نیز چنین است. یک فضای تغییرجا تحویل‌ناپذیر همیشه دارای گرافی تحویل‌ناپذیر است که پوششی برای آن فضا است. البته چنین فضایی دارای پوشش‌های تحویل‌پذیر نیز است. اما همیشه دارای پوشش تحویل‌ناپذیر هم است.

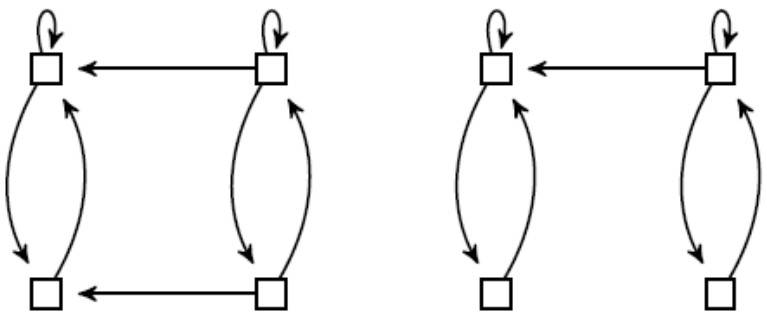
در شکل ۴ از سمت راست به‌ترتیب نمایش گراف‌های تحویل‌پذیر و تحویل‌ناپذیر هستند.

تعریف: *کلمه‌ی $w \in \mathcal{B}(X_{\mathcal{G}})$ یک کلمه‌ی جادویی[16] برای $\mathcal{G} = (G, \mathcal{L})$ است هرگاه هر مسیر در G با برچسب w به رأس یکسانی ختم شود.*

در شکل ۵ بلوک ۱ و ۲ جادویی هستند زیرا همه مسیرها با برچسب ۱ پایان یکسانی دارند. این مطلب برای بلوک ۲ نیز برقرار است اما برای بلوک ۰ معتبر نیست.

شکل ۴

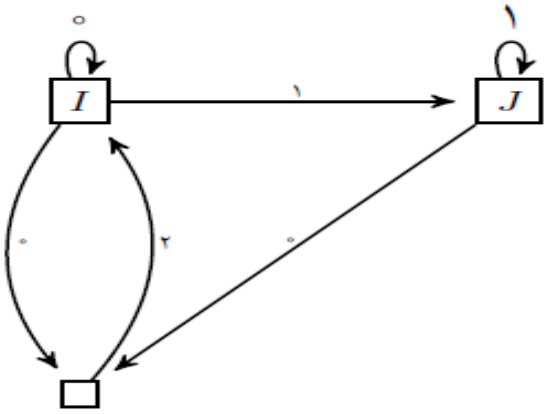
از سمت راست به‌ترتیب نمایش گراف‌های تحویل‌پذیر و تحویل‌ناپذیر هستند.



[16] *magic*

شکل ۵

بلوک ۱ و ۲ جادویی است ولی بلوک ۰ چنین نیست.



تعریف: *فضای تغییرجا X را سافیک می‌نامیم هرگاه $X = X_{\mathcal{G}}$ که $\mathcal{G}$ گرافی برچسب‌دار است. در این‌صورت $\mathcal{G}$ یک پوشش[۱۷] یا نمایش[۱۸] نامیده می‌شود.*

فضاهای تغییرجایی وجود دارند که سافیک نیستند. برای نمونه، مثال زیر را در نظر می‌گیریم.

مثال: *فضای X را روی الفبای $\mathcal{A} = \{a, b, c\}$ با مجموعه‌ی کلمات غیرمجاز*

$$\mathcal{F} = \{ab^m c^n a : m \neq n\}$$

در نظر می‌گیریم. این فضا، *تغییرجا متن آزاد* context free shift نامیده می‌شود.

نخست یادآوری می‌کنیم که برای هر بلوک دلخواه مانند m از فضای تغییرجا X مجموعه پیشرو $F_+(m)$ به‌صورت زیر تعریف می‌شود.

$$F_+(m) = \{u \in W(X) : mu \in W(X)\}.$$

چون برای $m \in \mathbb{N}$، $c^n a \in F_+(ab^m)$ اگروتنهااگر $n = m$، پس مجموعه‌های پیرو $F_+(ab^m)$ از یکدیگر مجزا هستند. بنابراین، تعداد مجموعه‌های پیرو نامتناهی است و در نتیجه X سافیک نیست [۳].

پوشش‌های یک تغییرجا سافیک لزوماً یکتا نیستند. حال پوششی را معرفی می‌کنیم که هم از نظر تئوری و هم عملی اهمیت بیشتری دارد.

تعریف: *گراف برچسب‌دار $\mathcal{G}$ از راست-معین[۱۹] نامیده می‌شود هرگاه برای هر رأس I، یال‌های خروجی از I دارای برچسب‌های متفاوت باشند.*

در گراف‌های از راست-معین، برای هر کلمه‌ی w و رأس I، حداکثر یک مسیر با ابتدای I و برچسب w وجود دارد. این گراف‌ها، در علوم کامپیوتر قطعی[۲۰] نامیده می‌شوند و کاربردهای متنوعی در کد کننده‌ها[۲۱] و باز کننده‌های کد[۲۲] دارند.

---

[17] *cover*
[18] *presentation*
[19] *right-resolving*
[20] deterministic
[21] encoders
[22] decoders

در شکل ۳، گراف G_2 از راست معین نیست. اما گراف G_3 از راست معین است.

در گراف‌های از چپ-معین یال‌های ورودی به هر رأس دارای برچسب‌های متمایز هستند. گرافی را که هم از راست و هم از چپ-معین باشد، از دو طرف-معین می‌نامیم.

در شکل ۳، گراف G_2 از چپ معین است. اما گراف G_3 چنین نیست. همچنین گراف G_1 از دوطرف معین است.

تعریف: *گراف برچسب‌دار $\mathcal{G} = (G, \mathcal{L})$ را در نظر می‌گیریم. برای $I \in \mathcal{V}$، مجموعه‌ی پیرو $F_{\mathcal{G}}(I)$ در $\mathcal{G}$، خانواده‌ی برچسب همه‌ی مسیرهایی با ابتدای I است؛ یعنی،*

$$F_{\mathcal{G}}(I) = \{\mathcal{L}(\pi): \pi \in \mathcal{B}(X_G) \text{ و } i(\pi) = I\}.$$

گراف $\mathcal{G}$ پیرو منفک[23] است هرگاه رئوس مختلف دارای مجموعه‌های پیرو متمایز باشند؛ یعنی، اگر $I \neq J$، آنگاه $F_{\mathcal{G}}(I) \neq F_{\mathcal{G}}(J)$.

در گراف $\mathcal{G} = (G, \mathcal{L})$، دو رأس I و J را هم‌ارز گوییم هرگاه دارای مجموعه‌های پیرو یکسان باشند؛ یعنی، $F_{\mathcal{G}}(I) = F_{\mathcal{G}}(J)$. رده‌های هم‌ارزی مجموعه‌ی رئوس $\mathcal{G}$ به‌صورت $\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_r$ هستند.

رمزنگاری

در این بخش تعاریف مقدماتی و قضایای لازم برای رمزنگاری بیان خواهند شد. تعاریف و قضایا از مرجع [۴] آورده شده‌اند اما توابع تعریف شده برای رمزنگاری، تفکیک متن، مثال‌ها و تعمیم روش رمزنگاری از ایده‌های نویسنده هستند.

تعریف: *گراف برچسب‌دار $\mathcal{G}$ از راست بسته است اگر هر دو مسیر با طول $D + 1$ که رأس آغازی و برچسب یکسانی دارند، دارای یال آغازی یکسان باشند. دراین‌حالت گوییم گراف G دارای تاخیر D است.*

گراف‌های از راست بسته تعمیم طبیعی گراف‌های از راست معین هستند. درواقع یک گراف برچسب‌دار از راست معین است اگر و تنها اگر از راست بسته و با تاخیر ۰ باشد.

یک گراف برچسب‌دار از چپ بسته با تاخیر D است هرگاه هر دو مسیر با طول $D + 1$ و برچسب یکسان که به رأس یکسانی ختم می‌شوند، دارای یال‌های انتهایی یکسانی باشند.

تعریف: *برچسب $\mathcal{J}: G \longrightarrow \mathcal{A}$ را رنگ‌آمیزی شده[24] برای گراف G گوییم، هرگاه به ازای هر رأس I از گراف G تناظر یک به یکی بین عناصر $\mathcal{A}$ و برچسب یال‌های خروجی از رأس I باشد.*

تعریف: *فرض کنیم $\mathcal{J}$ و $\mathcal{O}$ به‌ترتیب برچسب‌های رنگ‌آمیزی شده و از راست بسته گراف G باشند. دراین‌صورت سه‌تایی $(G, \mathcal{J}, \mathcal{O})$ یک کد متناهی نامیده می‌شود.*

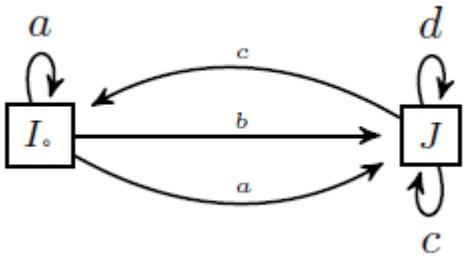
اگر درجه‌ی خروجی گراف G ثابت n باشد یعنی از هر رأس آن n یال خارج شود. همچنین X فضای تغییرجایی باشد که $\mathcal{O}(X) \subset X$، آنگاه $(G, \mathcal{J}, \mathcal{O})$ یک (X, n) کد متناهی نامیده می‌شود.

نگاشت $\mathcal{J}$ را نگاشت برچسب ورودی و $\mathcal{O}$ را نگاشت برچسب خروجی گوییم.

[23] follower-separated
[24] *coloring road*

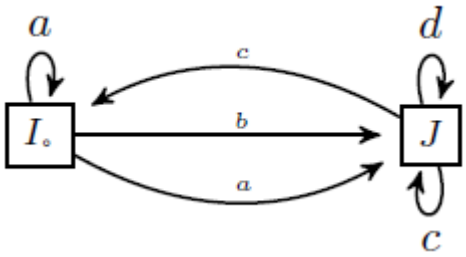
شکل ۶

نمایشی برای یک گراف از راست بسته.



شکل ۷

نمایش برچسب ورودی.



مثال: *فرض کنیم G گراف شکل ۶ باشد. گراف G از راست بسته و درجه خروجی ثابت ۳ دارد. بنابراین برچسب خروجی مانند $\mathcal{O}$ برای یک کدمتناهی فراهم می‌کند. گراف شکل ۷ یک انتخاب برای برچسب ورودی $\mathcal{J}$ است.*

از این نگاشت‌ها برای تبدیل دنباله‌هایی از یک $n-$تغییرجا کامل به دنباله‌هایی از فضای تغییرجا X استفاده می‌کنیم. بدین‌منظور فرض کنیم $(G,\mathcal{J},\mathcal{O})$ یک $(X,n)-$کد متناهی باشد. رأس $I_.$ از گراف G را ثابت درنظر می‌گیریم. فرض کنیم

$$x:=x_0x_1x_2\ldots\in\{0,1,\ldots,n-1\}^{\mathbb{N}}$$

دراین‌صورت مسیر یکتای $e_.e_1\ldots$ در گراف G وجود دارد که $\mathcal{J}(e_.e_1\ldots)=x_.x_1\ldots$ و $i(e_.e_1e_2\ldots)=I_.$ [۵]. حال نگاشت $\mathcal{O}$ را بر این مسیر اثر می‌دهیم تا خروجی

$$y_+:=\mathcal{O}(e_0)\mathcal{O}(e_1)\mathcal{O}(e_2)\ldots\in X^+$$

را به‌دست آوریم. به‌طور طبیعی عمل فوق برگشت‌پذیر است یعنی ما می‌توانیم ورودی x_+ را از خروجی y_+ به‌دست آوریم.

قضیه‌ی بعد شرط لازم و کافی برای اینکه یک کد متناهی وجود داشته باشد را بیان می‌کند.

قضیه: *فرض کنیم X یک فضای سافیک و $n>1$ باشد. دراین‌صورت یک $(X,n)-$کد متناهی موجود است اگر و تنها اگر $h(X)>\log n$.* [۴]

حال بررسی می‌کنیم چنان‌چه شرط آنتروپی در قضیه‌ی بالا برقرار نباشد چگونه از آن قضیه استفاده نماییم؟

نخست عدد گویای $\frac{p}{q}$ را چنان انتخاب می‌کنیم که

$$h(X) > \frac{p}{q}.$$

پس $h(X^q) > \log_2 p$ درنتیجه بنا به قضیه ۴.۵، یک $-(X^q, 2^p)$ کد متناهی موجود است. این کد متناهی هر بلوک به طول p از تغییرجا کامل $\{0,1\}^{\mathbb{Z}}$ را به یک بلوک q تایی از X تبدیل می‌کند.

مثال: *فرض کنیم G گراف شکل ۴.۶ باشد. قرار می‌دهیم $X = X_G$ دراین‌صورت آنتروپی فضای تغییرجای X تقریباً برابر با ۰.۵۵ است پس $p = 1$ و $q = 2$ را انتخاب می‌کنیم. درنتیجه الفبای ورودی ۰ و ۱ هستند و الفبای خروجی بلوک‌های به طول ۲ از فضای تغییرجا X یعنی $\{00,01,10\}$ هستند.*

شکل ۸

نمایش گرافی که شرط آنتروپی برای رمزنگاری در آن برقرار نیست.

رمزنگاری

در این قسمت می‌خواهیم گراف‌ها و نگاشت‌های برچسب ورودی و خروجی را معرفی نماییم تا بتوانیم هر متن دلخواهی را رمزنگاری نماییم. درواقع برای آنکه رمزنگاری پیچیده‌تر شود، ما از دو گراف، دو برچسب ورودی و دو برچسب خروجی استفاده می‌کنیم. در این روش وقتی متن را به رایانه می‌دهیم تا رمزنگاری کند، رایانه قسمتی از متن را به کمک برچسب ورودی اولی رمزنگاری می‌کند سپس قسمت دیگری را به کمک برچسب ورودی دومی رمزنگاری می‌کند و برای ادامه رمزنگاری به سراغ نگاشت برچسب ورودی اولی می‌رود و مجدد بعد از پیمودن طول از قبل تعیین شده‌ای، نگاشت برچسب ورودی را تغییر می‌دهد و این عملیات را تا پایان ادامه می‌دهد.

کوچکترین واحد ذخیره‌سازی در رایانه‌ها بیت است و هر بیت یک مقدار دودویی دارد یعنی می‌تواند ۱ یا ۰ را در خود جای دهد. همچنین واحد بزرگتر از بیت، بایت است. یک بایت شامل هشت بیت است، به این معنی که می‌تواند $2^8 = 256$ مقادیر مختلف داشته باشد. ازطرفی تمام ارتباطات و فعالیت‌های رایانه مانند ذخیره‌سازی داده‌ها و حتی محاسبات بر اساس اعداد ۰ و ۱ یا همان زبان دودویی انجام می‌شوند. درنتیجه بافرض آنکه W_8 مجموعه‌ی همه‌ی هشت‌تایی‌های مرتب از عناصر ۰ و ۱ باشد، آنگاه تعداد عناصر مجموعه W_8 برابر 2^8 است.

درنتیجه هر داده را می‌توان نظیر یک رشته از عناصر مجموعه‌ی W_8 دانست. پس هر متن توسط عناصر W_8 تولید می‌شود.

اکنون به هر عنصر مجموعه W_8 دو برچسب می‌دهیم. به عبارتی به ازای هر $0 < i < 257$ عناصر a_i و b_i را از W_8 طوری انتخاب می‌کنیم که برای هر $i > 1$، $a_i \notin \{a_1, a_2, \dots, a_{i-1}\}$ و $b_i \notin \{b_1, b_2, \dots, b_{i-1}\}$ درواقع داریم

$$W_8 = \{a_1, a_2, \dots, a_{256}\} = \{b_1, b_2, \dots, b_{256}\}.$$

از بلوک‌های مجموعه‌ی اولی برای نمایش داده‌ها و از مجموعه‌ی دوم برای رمزنگاری استفاده خواهیم کرد. بدین‌منظور نخست توابعی را معرفی می‌کنیم که بوسیله‌ی آنها داده‌ی $a_{i_1} a_{i_2} \dots a_{i_n}$ را به بلوک رمزی $b_{i_1} b_{i_2} \dots b_{i_n}$ تبدیل می‌کند.

مجموعه G_1 و توابع $\mathcal{O}_1, \mathcal{J}_1 : G_1 \longrightarrow W_8$ را به‌صورت زیرتعریف می‌کنیم:

$$G_1 = \{(i,0,0)\}_{i=1}^{128} \cup (i,0,1)\}_{i=129}^{256} \cup (i,0,2)\}_{i=257}^{384} \cup (i,0,3)\}_{i=385}^{512}.$$

$$\mathcal{J}_1(i,0,0) = \mathcal{J}_1(i,0,1) = a_i, \mathcal{J}_1(i,1,0) = \mathcal{J}_1(i,1,1) = a_{i-256}.$$

$$\mathcal{O}_1(i,0,0) = b_i, \mathcal{O}_1(i,0,1) = b_{i-128}, \mathcal{O}_1(i,1,0) = b_{i-256}$$

و

$$\mathcal{O}_1(i,1,1) = b_{i-128}.$$

برای تعریف کردن تابع $\mathcal{J}_1^{-1}$ فرض کنیم $n > 2$ و $a_{i_1}, a_{i_2}, \dots, a_{i_n} \in W_8$ دلخواه باشند. اعداد $r_1, r_2, \dots, r_n$ و $k_2, k_3, \dots, k_n$ موجودند که

$$\mathcal{J}_1(i_1,0,r_1) = a_{i_1}, \mathcal{J}_1(k_2,r_1,r_2) = a_{i_2}, \dots, \mathcal{J}_1(k_n,r_{n-1},r_n) = a_{i-n}.$$

حال تعریف می‌کنیم:

$$\mathcal{J}_1^{-1}(a_{i_1}a_{i_2}\dots a_{i_n}) = (i_1,0,r_1)(k_2,r_1,r_2)\dots(k_n,r_{n-1},r_n).$$

دراین‌صورت ترکیب توابع $\mathcal{O}_1$ و $\mathcal{J}_1^{-1}$ به‌صورت زیر تعریف می‌شوند:

$$\mathcal{O}_1\mathcal{J}_1^{-1}(a_{i_1}a_{i_2}\dots a_{i_n}) = \mathcal{O}_1(i_1,0,r_1)\mathcal{O}_1(k_2,r_1,r_2)\dots\mathcal{O}_1(k_n,r_{n-1},r_n).$$

چون برد $\mathcal{O}_1$ تابع برحسب بلوک‌های b_i می‌باشد، پس درواقع تابع ترکیب $\mathcal{O}_1\mathcal{J}_1^{-1}$ به هر داده یک کد نسبت می‌دهد. درواقع دامنه توابع $\mathcal{O}_1$ و $\mathcal{J}_1$ یال‌های گراف G_1 می‌باشند. یعنی به هر یال گراف G_1 یک برچسب می‌دهند. حال مجموعه G_2 و نگاشت‌های برچسب ورودی و خروجی $\mathcal{O}_2, \mathcal{J}_2 : G_2 \longrightarrow W_8$ را به‌صورت زیر تعریف می‌کنیم:

$$G_2 = G_2^1 \cup G_2^2 \cup G_2^3 \cup G_2^4$$

جایی‌که

$$G_2^1 = \{(i,0,0)\}_{i=1}^{64} \cup (i,0,1)\}_{i=65}^{128} \cup (i,0,2)\}_{i=129}^{192} \cup (i,0,3)\}_{i=193}^{256}$$

$$G_2^2 = \{(i,1,0)\}_{i=257}^{320} \cup (i,1,1)\}_{i=321}^{384} \cup (i,1,2)\}_{i=385}^{448} \cup (i,1,3)\}_{i=449}^{512}$$

$$G_2^3 = \{(i,2,0)\}_{i=513}^{576} \cup (i,2,1)\}_{i=577}^{640} \cup (i,2,2)\}_{i=641}^{704} \cup (i,023)\}_{i=705}^{768}$$

$$G_2^4 = \{(i,3,0)\}_{i=769}^{832} \cup (i,3,1)\}_{i=833}^{896} \cup (i,3,2)\}_{i=897}^{960} \cup (i,3,3)\}_{i=961}^{1024}.$$

$$\mathcal{J}_2(i,j,k) = \begin{cases} a_i & j = 0, k = 1,2,3 \\ a_{i-256} & j = 1, k = 1,2,3 \\ a_{i-512} & j = 2, k = 1,2,3 \\ a_{i-796} & j = 3, k = 1,2,3 \end{cases}$$

$$\mathcal{O}_2(i,j,k)=\begin{cases} b_i & j=k=0 \\ b_{i-64} & j=0,k=1 \\ b_{i-128} & j=0,k=2 \\ b_{i-192} & j=0,k=3 \\ b_{i-192} & j=1,k=0 \\ b_{i-256} & j=1,k=1 \\ b_{i-320} & j=1,k=2 \\ b_{i-384} & j=1,k=3 \\ b_{i-384} & j=2,k=0 \\ b_{i-448} & j=2,k=1 \\ b_{i-512} & j=2,k=2 \\ b_{i-576} & j=2,k=3 \\ b_{i-576} & j=3,k=0 \\ b_{i-640} & j=3,k=1 \\ b_{i-704} & j=3,k=2 \\ b_{i-768} & j=3,k=3 \end{cases}$$

برای تعریف کردن تابع $\mathcal{J}_2^{-1}$، فرض کنیم $n>2$ و $a_{i_1},a_{i_2},\dots,a_{i_n}\in W_8$ دلخواه باشند. اعداد $r_1,r_2,\dots,r_n$ و $k_2,\dots,k_n$ موجودند که

$$\mathcal{J}_2(i_1,0,r_1)=a_{i_1},\mathcal{J}_2(k_2,r_1,r_2)=a_{i_2},\dots,\mathcal{J}_2(k_n,r_{n-1},r_n)=a_{i-n}.$$

حال تعریف می‌کنیم

$$\mathcal{J}_2^{-1}(a_{i_1}a_{i_2}\dots a_{i_n})=(i_1,0,r_1)(k_2,r_1,r_2)\dots(k_n,r_{n-1},r_n).$$

دراین‌صورت ترکیب توابع $\mathcal{O}_2$ و $\mathcal{J}_2^{-1}$ به‌صورت زیر تعریف می‌شود:

$$\mathcal{O}_2\mathcal{J}_2^{-1}(a_{i_1}a_{i_2}\dots a_{i_n})=\mathcal{O}_2(i_1,0,r_1)\mathcal{O}_2(k_2,r_1,r_2)\dots\mathcal{O}_2(k_n,r_{n-1},r_n).$$

همانند ترکیب $\mathcal{O}_1\mathcal{J}_1^{-1}$، ترکیب $\mathcal{O}_2\mathcal{J}_2^{-1}$ نیز عمل رمزنگاری را انجام می‌دهد. درواقع هدف ما آن است که یک داده را ابتدا افراز کنیم سپس عضوهای آن افراز را توسط توابع $\mathcal{O}_1\mathcal{J}_1^{-1}$ و $\mathcal{O}_2\mathcal{J}_2^{-1}$ رمزنگاری کنیم.

اکنون تمامی توابع لازم برای رمزنگاری و رمزگشایی تعریف شده‌اند. فرض کنیم $a_{i_1}a_{i_2}\dots a_{i_n}$ یک بلوک n تایی باشد. چون طول هر a_i برابر ۸ است پس داده‌ی ورودی بلوکی به طول $N:=8n$ از عناصر ۰ و ۱ می‌باشد. نخست مجموعه‌ی $\{n_1,n_2,\dots,n_k\}$ که آن را مجموعه‌ی افراز نامیم را چنان انتخاب می‌کنیم که $n_1+n_2+\dots+n_k=N$ و هر n_i بر ۸ بخشپذیر باشد.

درواقع مجموعه افراز تعدادی کلید است. عدد k یک کلید خصوصی و عناصر افراز کلیدهای عمومی هستند.

حال بلوک داده شده را به k زیربلوک $A_1,A_2,\dots,A_k$ تقسیم می‌کنیم به‌قسمی‌که طول هر A_i برابر n_i بوده و

$$a_{i_1}a_{i_2}\dots a_{i_n}=A_1A_2\dots A_k.$$

(برای نمونه با انتخاب مجموعه افراز $\{24,40,16\}$، بلوک داده شده‌ی $a_1a_2\dots a_{10}$ به‌صورت $a_1a_2a_3$ ، $a_4a_5a_6a_7a_8$ و a_9a_{10} تقسیم می‌شود.) سپس به هر عضو افراز بجز عضو آخر، اولین بلوک ۸ تایی عضو بعدی افراز را به آن اضافه می‌کنیم و به عضو آخر افراز، بلوک a_{i_n} را اضافه می‌کنیم. (برای نمونه با انجام این عمل بلوک نمونه‌ی قبلی به‌صورت زیر تبدیل می‌شود.

$$a_1a_2a_3a_4a_4a_5a_6a_7a_8a_9a_9a_{10}a_{10}.)$$

حال توابع $\mathcal{O}_1\mathcal{J}_1^{-1}$ و $\mathcal{O}_2\mathcal{J}_2^{-1}$ به ترتیب و یک در میان روی بلوک‌های افراز شده‌ی جدید اثر می‌دهیم که بلوک کدشده‌ی $b_{j_1}b_{j_2}\dots b_{j_{n+k}}$ به‌دست می‌آید. (در نمونه‌ی بیان شده عمل توابع مذکور ما را به بلوکی مانند $b_1\dots b_4b_5\dots b_{10}.b_{11}\dots b_{13}$ می‌رساند.)

بازخوانی

در این قسمت می‌خواهیم عکس رمزنگاری را انجام دهیم و یک بلوک کدشده را بازخوانی کنیم یعنی آن را به متن اولیه تبدیل کنیم. برای تعریف کردن تابع $\mathcal{O}_1^{-1}$ فرض کنیم $n > 2$ و $a_{j_1}, a_{j_2}, \dots, a_{j_n} \in W_8$ دلخواه باشند. اعداد $r_1, r_2, \dots r_{n-1}$ و $k_1, k_2, \dots k_{n-1}$ موجودند که

$$\mathcal{O}_1(k_1, 0, r_1) = b_{j_1}, \mathcal{O}_1(k_2, r_1, r_2) = b_{j_2}, \dots, \mathcal{O}_1(k_n, r_{n-1}, r_n) = b_{j_n}.$$

حال تعریف می‌کنیم:

$$\mathcal{O}_1^{-1}(b_{j_1} b_{j_2} \dots b_{j_n}) = (k_1, 0, r_1)(k_2, r_1, r_2) \dots (k_n, r_{n-1}, r_n).$$

دراین‌صورت ترکیب توابع $\mathcal{O}_1^{-1}$ و $\mathcal{J}_1$ به‌صورت زیر تعریف می‌شوند:

$$\mathcal{J}_1\mathcal{O}_1^{-1}(b_{j_1} b_{j_2} \dots b_{j_n}) = \mathcal{J}_1(k_1, 0, r_1)\mathcal{J}_1(k_2, r_1, r_2) \dots J_1(k_n, r_{n-1}, r_n).$$

به‌طریق مشابه $\mathcal{O}_2^{-1}$ و $\mathcal{J}_2\mathcal{O}_2^{-1}$ تعریف می‌شوند.

به کمک دو ترکیب تعریف شده‌ی قبلی می‌توانیم یک کد را بازخوانی کنیم. بدین منظور فرض کنیم $b_{j_1} b_{j_2} \dots b_{j_{n+k}}$ داده شده باشد. بلوک داده شده را به k زیربلوک $B_1, B_2, \dots, B_k$ تقسیم می‌کنیم به‌قسمی‌که طول B_i برابر با $n + 8$ بوده و

$$b_{j_1} b_{j_2} \dots b_{j_{n+k}} = B_1 B_2 \dots B_k.$$

حال عمل توابع $\mathcal{J}_1\mathcal{O}_1^{-1}$ و $\mathcal{J}_2\mathcal{O}_2^{-1}$ به ترتیب و یک درمیان روی بلوک‌های افراز شده یعنی $B_1, B_2, \dots, B_k$ متن اولیه‌ی $a_{i_1} a_{i_2} \dots a_{i_n}$ را نتیجه می‌دهد.

نتیجه‌گیری

در این مقاله تعداد کلیدهای عمومی از کلید خصوصی به‌دست می‌آیند. البته این بدان معنی نیست که با داشتن یکی از کلیدها می‌توان به دیگر کلیدها دسترسی پیدا کرد. زیرا کلید خصوصی تنها تعداد کلیدهای عمومی را مشخص می‌کند. موارد زیر تعدادی از دلایل متمایز بودن این روش رمزنگاری را با دیگر روش‌ها مشخص می‌کنند.

۱. تعداد کلیدها بیش از دو تا است.

۲. انتخاب تعداد کلیدها در اختیار کاربر است و می‌توان تعداد آن‌ها را به هر تعداد دلخواه افزایش داد.

۳. با تغییر متن کلیدهای خصوصی و عمومی تغییر می‌کنند.

۴. در یک متن می‌توان بارها کلیدهای خصوصی و عمومی را تغییر داد.

به‌سادگی می‌توان این روش را تعمیم داد و رمزنگاری پیچیده‌ای به‌وجود آورد. بدین‌منظور علاوه بر آنکه می‌توان از تعداد دلخواه توابع استفاده کرد، می‌توان به گونه‌ای ادامه داد تا رایانه موقع انتخاب، برحسب قاعده‌ای از قبل تعریف شده تابعی را انتخاب نماید. این عمل را علاوه بر افزایش تعداد توابع، می‌توان با انتخاب توابع برحسب متن نیز به‌دست آورد. افزایش تعداد عناصر افراز یا هم‌پوشانی داده‌ها نیز از جمله راه‌هایی برای افزایش ایمنی هستند.

تعارض منافع

در انجام مطالعه حاضر، هیچگونه تضاد منافعی وجود ندارد.

مشارکت نویسندگان

در نگارش این مقاله تمامی نویسندگان نقش یکسانی ایفا کردند.

موازین اخلاقی

در انجام این پژوهش تمامی موازین و اصول اخلاقی رعایت گردیده است.

شفافیت داده‌ها

داده‌ها و مآخذ پژوهش حاضر در صورت درخواست از نویسنده مسئول و ضمن رعایت اصول کپی رایت ارسال خواهد شد.

حامی مالی

این پژوهش حامی مالی نداشته است

منابع

[1] P. Amudha and A.C. Charles Sagayaraj and A.C.Shantha Sheela, An Application of Graph Theory in Cryptography, Volume 119 No. 132018,375 − 383, International Journal of Pure and Applied Mathematics.

[2] W. Etaiwi, Encryption Algorithm Using Graph Theory, January 2014, Journal of Scientific Research and Reports, 3(19): 2519 − 2527.

[3] D. Fiebig and U. Fiebig, Covers for coded systems, Contemporary Mathematics, 135, 1992, 139 − 179.

[4] D. Lind and B. Marcus, An introduction to symbolic dynamics and coding, Cambridge Univ. Press. 2005.

[5] K. Thomsen, On the ergodic theory of synchronized systems, Ergod. Th. Dynam. Sys. 356(2016)1235 − 1256.

[6] K. Thomsen, On the structure of a sofic shift space, American Mathematical Society, 356, Number 9(2014),3557 − 3619.