International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 04 | April - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

# False-Bottom Encryption: Deniable Encryption from Secret Sharing

**Suyash Agrawal[#], Aleti Navaneetha[*], Pola Saisindhu[*], Salla Laxmi Prasanna[*]**

[#]Assistant Professor, Department of IT, Guru Nanak Institutions Technical Campus, Hyderabad.
[*]Department of IT, Guru Nanak Institutions Technical Campus, Hyderabad.

**Abstract:** We show how to implement a deniable encryption method from secret sharing. Unlike the related concept of honey encryption, which employs a preprocessing step in symmetric encryption to re-shape the distribution of a plaintext towards making the real plaintext indistinguishable from a ciphertext for a fake message, we can avoid both, computational intractability assumptions and preprocessing of the data. This accomplishes deniability against an attacker that can force decryptions, and it can brute-force break a ciphertext with sufficient computational power. Following the concept of plausible deniability, we herein have different decryption keys to open up distinct plaintexts from within the same ciphertext. For instance, a plaintext revealed from a ciphertext with a key which was shared by a victim under duress, will convince the attacker that it is real, while the actual secret remains unnoticed. False Bottom Encryption constructs a symmetric scheme (in the sense of using the same key to encrypt and decrypt) that shares the properties of both honey encryption and deniable encryption. We specifically formalize and differentiate ‘‘deniable’’ from ‘‘plausibly deniable’’ as a security feature, showing how plausible deniability falls back to (only) deniability, depending on the plaintext distribution. Our scheme is simple, lightweight to implement and efficient in terms of encryption and decryption, and is based on secret sharing. As such, we do not rely on computational intractability. We corroborate the construction by giving numeric examples and providing implementations of the method as a Jupyter notebook supplementary to this work.

**Keywords:** Deniable encryption, honey encryption, plausible deniability, secret sharing

## I.INTRODUCTION

Generally speaking, a sender and receiver must first share information to communicate safely. In many cases, encryption and good password protection may be sufficient to safeguard your data. For instance, with AES, the sender and receiver share the same key in a symmetric encryption system. However, using the RSA technique, the sender and receiver of an asymmetric-key encryption technique exchange a public system parameter and the recipient’s public key, with the public key delivery occurring through public key infrastructure. These general encryption techniques offer a security guarantee against eavesdropping attempts, but they fall short when faced with threats of coercion. Even if the attacker does not have access to the key, if it intercepts the ciphertext, it may be able to force both the sender and the receiver to decrypt the message. Non-committing encryption [1] and deniable encryption [2] have been presented as solutions to this issue. Users can decrypt an existing ciphertext associated with a certain counterfeit message using these two different encryption algorithms. The

first algorithm is called the sender's encryption algorithm to encrypt a message under a secret key sk. The second algorithm, known as the faking algorithm, is publicly known, and the sender uses this fake algorithm to produce fake messages. Getting the same ciphertext from two different algorithms is computationally cumbersome. In our work, we show how the actual message and the fake message ciphertexts can both be produced using a single algorithm that also is computationally efficient. There are many circumstances under which plausible deniability may also be necessary. If your opponents cannot obtain. your password, strong encryption can keep them out. However, if the threat model incorporates coercion, such as the prospect of a jail term or torture, you might give up and hand over the key to rescue yourself. As a result, the attacker would have access to the data, perhaps putting you at risk for later repercussions. The idea of plausible deniability originates in politics and espionage and refers to one's capacity to downplay one's culpability for, or knowledge of specific facts or events. It may entail carrying out operations in a way that leaves no trace, especially changing systems around particular people, to enable them to honestly deny their knowledge of what took place. Destruction of evidence is another method that can be used to make a given action plausible to deny, but there are also positive use-cases as we will outline next. We question whether it is conceivable to produce ciphertext that appears to be for certain claimed receivers but are actually for different receivers.

Imagine that Alice wants to secretly send her friend John a message. If she encrypts and sends it to John, she could be asked by her mother who the message was for and command her to decrypt the message. Consequently, John might get a call from Alice's mother, asking him to confirm as well what it says. To prevent this from happening, Alice can encrypt the message using deniable encryption. Alice will first prepare a pair of texts. One is a trivial message for Bob, whereas the other is a simple covert message for John. John's text is jointly encrypted with Bob's text by Alice using a suitable encryption algorithm. Alice posts the ciphertext to a public channel and requests her friends to download the message. The only two people who can successfully decrypt the ciphertext are Bob and John, but they produce two different messages: the fake message and the real message. The term ‘‘successful decryption’’ refers to the fact that Bob and John are able to decrypt and receive useful messages from the sender. Alice can tell her mother that the ciphertext is for Bob and reveal the message that was transmitted to Bob when questioned. Considering that Bob only knows what he has received, he can also be a trustworthy witness. Even if Alice's mother thinks something is concealed in the ciphertext, she cannot determine which of Alice's friends is the true recipient. In this case, Alice does not need to help John because her mother would not be able to suspect John, unless she suspects all of Alice's friends.

## II.LITERATURE SURVEY

J. Li, Q. Yu, and Y. Zhang, Attribute based encryption (ABE) is widely applied in cloud computing settings due to its fine-grained access control. Most ABE schemes do not consider the side channel attacks which may leak the secret information of cryptosystems. Leakage-resilient cryptography aims to model security for various side channel attacks. In this paper, we first give the formal definition and security model of hierarchical attribute based encryption (HABE) with continuous leakage-resilience. Furthermore, we present a ciphertext-policy

HABE scheme with continuous leakage-resilience. The proposed scheme is resilient to master key leakage and secret key leakage. We prove the security of our scheme under composite order bilinear group assumptions by using dual system encryption techniques. The performance of leakage-resilience is analyzed theoretically. If the depth of our proposed scheme is 1, the relative leakage ratio is almost up to 1/3. In addition, we give the performance comparison through experiments.

Hakan Ancin,Xi Chen, data privacy is critical in instilling trust and empowering the societal pacts of modern technology-driven democracies. Unfortunately, it is under continuous attack by overreaching or outright oppressive governments, including some of the world's oldest democracies. Increasingly-intrusive anti-encryption laws severely limit the ability of standard encryption to protect privacy. New defense mechanisms are needed. Plausible deniability (PD) is a powerful property, enabling users to hide the existence of sensitive information in a system under direct inspection by adversaries. Popular encrypted storage systems such as TrueCrypt and other research efforts have attempted to also provide plausible deniability. Unfortunately, these efforts have often operated under less well-defined assumptions and adversarial models. Careful analyses often uncover not only high overheads but also outright security compromise. Further, our understanding of adversaries, the underlying storage technologies, as well as the available plausible deniable solutions have evolved dramatically in the past two decades. The main goal of this work is to systematize this knowledge. It aims to:

1. identify key PD properties, requirements, and approaches;
2. present a direly-needed unified framework for evaluating security and performance;
3. explore the challenges arising from the critical interplay between PD and modern system layered stacks;
4. propose a new "trace-oriented" PD paradigm, able to decouple security guarantees from the underlying systems and thus ensure a higher level of flexibility and security independent of the technology stack.

This work is meant also as a trusted guide for system and security practitioners around the major challenges in understanding, designing, and implementing plausible deniability into new or existing systems.

P. Chi and C. Lei, cloud storage services have become increasingly popular. Because of the importance of privacy, many cloud storage encryption schemes have been proposed to protect data from those who do not have access. All such schemes assumed that cloud storage providers are safe and cannot be hacked; however, in practice, some authorities (i.e., coercers) may force cloud storage providers to reveal user secrets or confidential data on the cloud, thus altogether circumventing storage encryption schemes. In this paper, we present our design for a new cloud storage encryption scheme that enables cloud storage providers to create convincing fake user secrets to protect user privacy. Since coercers cannot tell if obtained secrets are true or not, the cloud storage providers ensure that user privacy is still securely protected.

C. Chen, A. Chakraborti, and R. Sion, encryption protects sensitive data from unauthorized access, yet is not sufficient when users are forced to surrender keys under duress. In contrast, plausible deniability enables users

to not only encrypt data but also deny its existence when challenged. Most existing plausible deniability work (e.g. the successful and unfortunately now-defunct TrueCrypt) tackles “single snapshot” adversaries, and cannot handle the more realistic scenario of adversaries gaining access to a device at multiple time points. Such “multi-snapshot” adversaries can simply observe modifications between snapshots and detect the existence of hidden data. Existing ideas handling “multi-snapshot” scenarios feature prohibitive overheads when deployed on practically-sized disks. This is mostly due to a lack of data locality inherent in certain standard access randomization mechanisms, one of the building blocks used to ensure plausible deniability. In this work, we show that such randomization is not necessary for strong plausible deniability. Instead, it can be replaced by a canonical form that permits most of writes to be done sequentially. This has two key advantages: 1) it reduces the impact of seek due to random accesses; 2) it reduces the overall number of physical blocks that need to be written for each logical write. As a result, PD-DM increases I/O throughput by orders of magnitude (10–100× in typical setups) over existing work while maintaining strong plausible deniability against multi-snapshot adversaries. Notably, PD-DM is the first plausible-deniable system getting within reach of the performance of standard encrypted volumes (dm-crypt) for random I/O.

Li, Y. Wang, Y. Zhang, and J. Han, attribute based encryption (ABE) is a popular cryptographic technology to protect the security of users' data. However, the decryption cost and cipher text size restrict the application of ABE in practice. For most existing ABE schemes, the decryption cost and cipher text size grow linearly with the complexity of access structure. This is undesirable to the devices with limited computing capability and storage space. Outsourced decryption is considered as a feasible method to reduce the user's decryption overhead, which enables a user to outsource a large number of decryption operations to the cloud service provider (CSP). However, outsourced decryption cannot guarantee the correctness of transformation done by the cloud, so it is necessary to check the correctness of outsourced decryption to ensure security for users' data. Current research mainly focuses on verifiability of outsourced decryption for the authorized users. It still remains a challenging issue that how to guarantee the correctness of outsourced decryption for unauthorized users. In this paper, we propose an ABE scheme with verifiable outsourced decryption (called full verifiability for outsourced decryption), which can simultaneously check the correctness for transformed ciphertext for the authorized users and unauthorized users. The proposed ABE scheme with verifiable outsourced decryption is proved to be selective CPA-secure in the standard model.

G. Gong and Q. Cha, outsourcing data to cloud servers, while increasing service availability and reducing users' burden of managing data, inevitably brings in new concerns such as data privacy, since the server may be honest-but-curious. To mediate the conflicts between data usability and data privacy in such a scenario, research of searchable encryption is of increasing interest. Motivated by the fact that a cloud server, besides its curiosity, may be selfish in order to save its computation and/or download bandwidth, in this paper, we investigate the searchable encryption problem in the presence of a semi-honest-but-curious server, which may execute only a fraction of search operations honestly and return a fraction of search outcome honestly. To fight against this strongest adversary ever, a verifiable SSE (VSSE) scheme is proposed to offer verifiable searchability in additional to the data privacy, both of which are further confirmed by our rigorous security analysis. Besides,

we treat the practicality/efficiency as a central requirement of a searchable encryption scheme. To demonstrate the lightweightness of our scheme, we implemented and tested the proposed VSSE on a laptop (serving as the server) and a mobile phone running Android 2.3.4 (serving as the end user). The experimental results optimistically suggest that the proposed scheme satisfies all of our design goals.

## III. RELATED WORK

The Generally, a sender and receiver must first share information to communicate safely. In many cases, encryption and good password protection may be sufficient to safeguard your data. For instance, with AES, the sender and receiver share the same key in a symmetric encryption system. the public key, with the public key delivery occurring through public key infrastructure. cipher text, it may be able to both the sender and the receiver to decrypt the data stored.

The scope of the project the construction of honey encryption makes use of distribution-transforming encoders that aim to shape the distribution of a random plaintext towards a desired and fixed target distribution. Our scheme can use such encoders as well, as a source of plausibly looking plaintexts to act as fakes. We will not make explicit use of such transformations, but mention them as a possible technical implementation of our assumption that fake plaintexts are producible with the same distribution as the real secret plaintexts.

### Demerits Observed in Existing works

- Users can decrypt an existing cipher text associated with a certain counterfeit message using these two different encryption algorithms.
- The first algorithm is called the sender's encryption algorithm to encrypt a message under a secret key sk.
- The second algorithm, known as the faking algorithm, is publicly known, and the sender uses this fake algorithm to produce fake messages. Getting the same cipher text from two different algorithms is computationally cumbersome.
- In our work, we show how the actual message and the fake message cipher texts can both be produced using a single algorithm that also is computationally efficient.
- However, using the RSA technique, the sender and receiver of an asymmetric-key encryption technique exchange a public system

## IV. PROPOSED SYSTEM

Unlike the related concept of honey encryption, which employs a preprocessing step in symmetric encryption to re-shape the distribution of a plaintext towards making the real plaintext indistinguishable from a cipher text data. For instance, with AES, the sender and receiver share the same key in a symmetric encryption system.

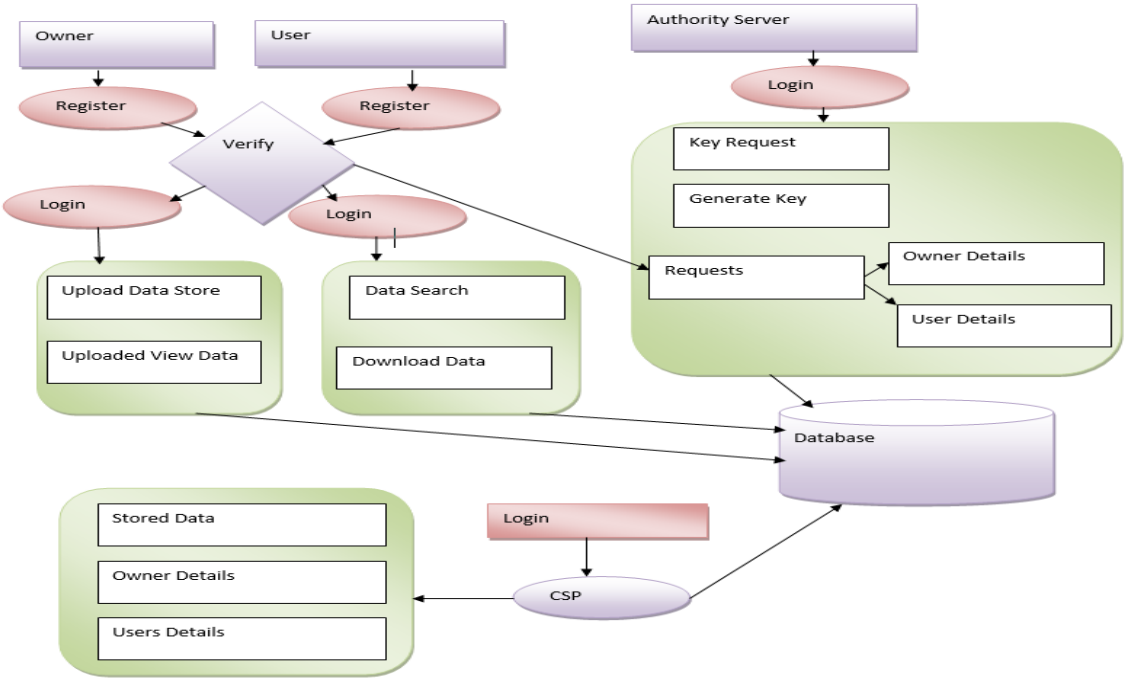


Figure 1. Architecture of the work.

A deniable shared key scheme and a public key scheme are two types of deniable encryption. A straight forward illustration of deniable encryption is the one-time pad: Let m be the original message to be encrypted data. n the area of symmetric encryption, ambiguous multi-symmetric cryptography has been proposed with goals similar to ours. They provide a discussion of various attack scenarios, including known-plaintext, chosen cipher text and others.

In the area of symmetric encryption, ambiguous multi-symmetric cryptography has been proposed with goals similar to ours. They provide a discussion of various attack scenarios, including known-plaintext, chosencipher text and others, all boiling down to the argument that this leaves the adversary with a linear system of equations that is under-determined. In basing their construction on numbertheoretic arguments, they nonetheless rely on computational intractability (of finding factorizations or primes, though not via a usual reductionist argument). The computational complexity of this prior construction is governed by Chinese remaindering. Our scheme improves over this in requiring only linear efforts (a number of field operations that is proportional to the number of inner plaintexts).

## V.MODULES OF THE WORK

### 1. User Interface Design

In this module we design the windows for the project. These windows are used for secure login for all users. To connect with server user must give their username and password then only they can able to connect the server.

If the user already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. Server will create the account for the entire user to maintain upload and download rate. Name will be set as user id. Logging in is usually used to enter a specific page.

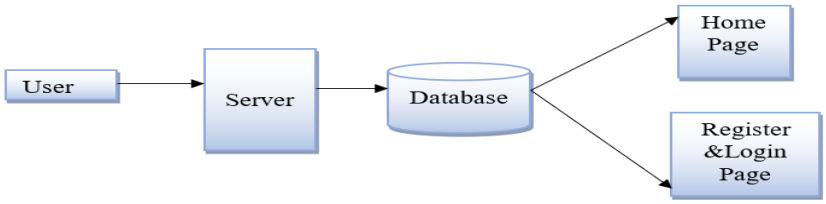


Figure 2. User interface

**2. Authority Server**

The first module is a authenticator server. First server has to login with a user id and password. Authority server has a key request. The authority server has a generate a key generator.The authority server has a requests. It will have a owner requests and it was also have a user requests from the database.

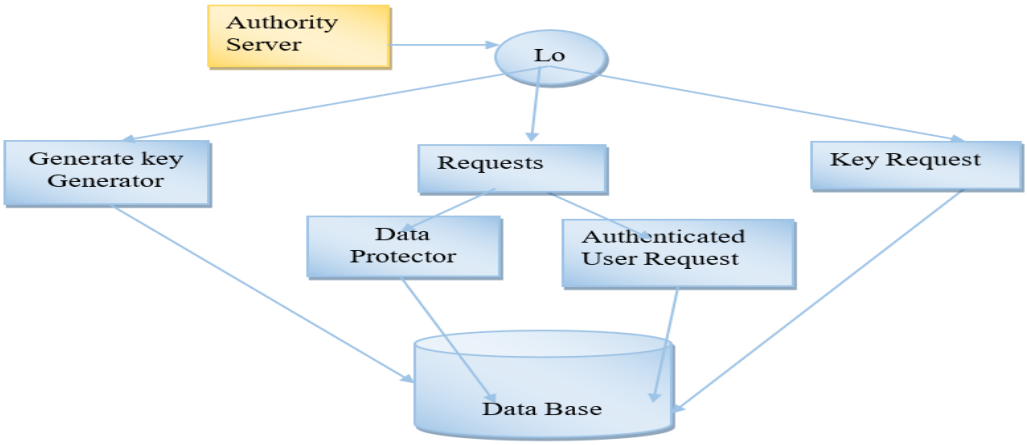


Figure 3. Authority server.

**3. Owner**

The third module. Owner has a register and login takes permission from the authority server. Authority server has a login with a user id and password. Owner has a uploaded data store. We can view a uploaded view data. Owner has a approve a data to share.

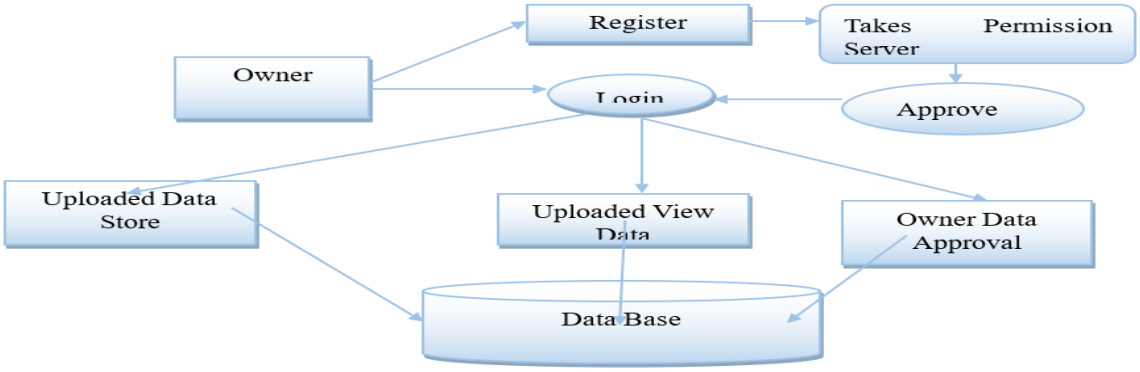


Figure 4. Owner details.

**4. User**

The fourth module is a user. User has a register with a user id and password. User has a to login takes permissions from the authority center. Authority server has to add a user then login. User has a data search a data.

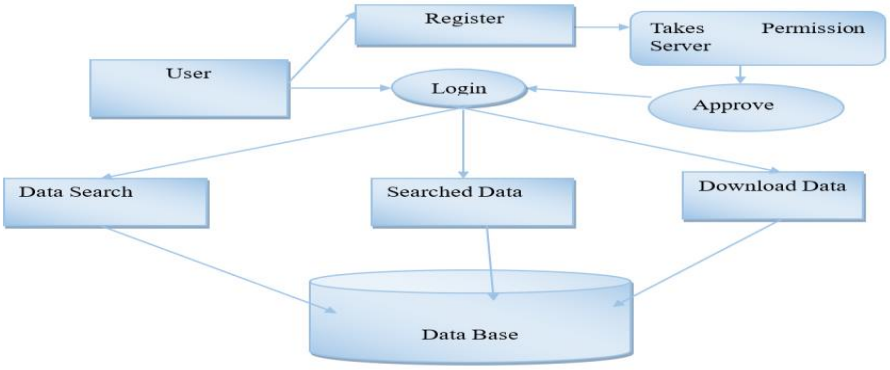


Figure 5. User details.

**5. CSP( Cloud Service Provider)**

Cloud has a fifth module.CSP has a login with a user id and password.CSP has a Stored data in the database.CSP has a details to have a owner details and it have a user details. The stores in a database.

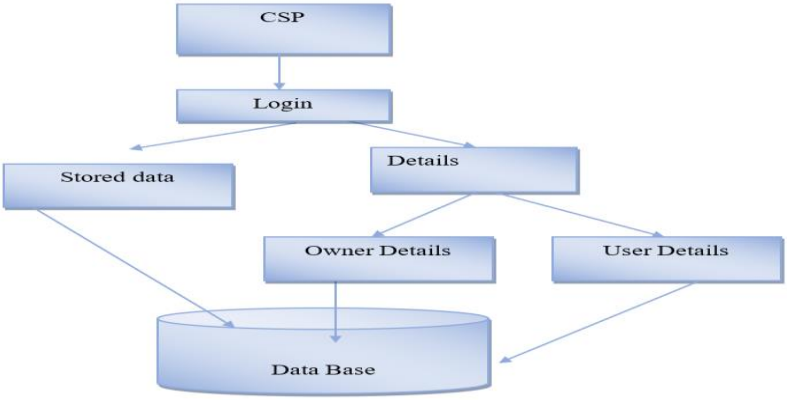


Figure 6. Cloud Service Provider.

## VI.PROPOSED ALGORITHM

### Honey Encryption

A Honey Encryption involves repeated decryption with random keys; this is equivalent to picking random plaintexts from the space of all possible plaintexts with a uniform distribution. This is effective because even though the attacker is equally likely to see any given plaintext, most plaintexts are extremely unlikely to be legitimate i.e. the distribution of legitimate plaintexts is non-uniform. Honey encryption defeats such attacks by first transforming the plaintext into a space such that the distribution of legitimate plaintexts is uniform. Thus an attacker guessing keys will see legitimate-looking plaintexts frequently and random-looking plaintexts infrequently. This makes it difficult to determine when the correct key has been guessed. In effect, honey encryption "[serves] up fake data in response to every incorrect guess of the data or encryption key.

The security of honey encryption relies on the fact that the probability of an attacker a plaintext to be legitimate can be calculated (by the encrypting party) at the encryption. This makes honey encryption difficult to apply in certain applications e.g. where the space of plaintexts is very large or the distribution of plaintexts is unknown. It also means that honey encryption can be vulnerable if this probability is miscalculated. For example, it is vulnerable to known-plaintext attacks: if the attacker has a crib that a plaintext must match to be legitimate, they will be able to brute-force even Honey Encrypted data if the encryption did not take the crib into account.

### 2.Hash Algorithm

There are majorly three components of hashing:

Key: A Key can be anything string or integer which is fed as input in the hash function the technique that determines an index or location for storage of an item in a data structure.
Hash Function: The hash function receives the input key and returns the index of an element in an array called a hash table. The index is known as the hash index.
Hash Table: Hash table is a data structure that keys to values using a special function called a hash function. Hash stores the data in an associative manner in an array where each data value has its own unique index.

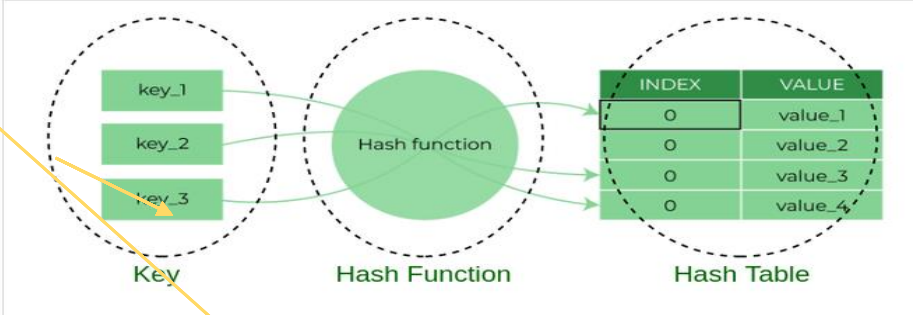


Figure 7 Hash algorithm Example.

## VII. SYMMETRIC ALGORITHM USED

Symmetric-key algorithms are algorithms for cryptography that use the same cryptographic keys for both the encryption of plaintext and the decryption of cipher text. The keys may be identical, or there may be a simple transformation to go between the two keys.[1] The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link.[2] The requirement that both parties have access to the secret key is one of the main drawbacks of symmetric-key encryption, in comparison to public-key encryption (also known as asymmetric-key encryption).[3][4] However, symmetric-key encryption algorithms are usually better for bulk encryption. With exception of the one-time pad they have a smaller key size, which means less storage space and faster transmission. Due to this, asymmetric-key encryption is often used to exchange the secret key for symmetric-key encryption.

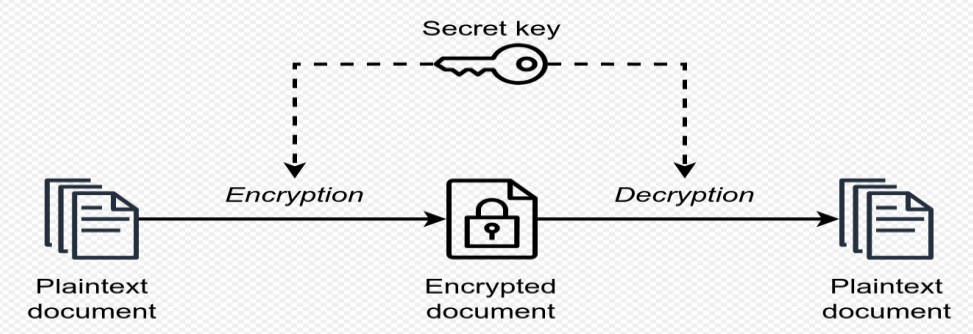


Figure 8. Symmetric key algorithm.

Attribute-based encryption is a generalization of public-key encryption which enables fine grained access control of encrypted data using authorization policies. The secret key of a user and the cipher text are dependent upon attributes (e.g. their email address, the country in which they live, or the kind of subscription they have). In such a system, the decryption of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text.[1]

A crucial security aspect of attribute-based encryption is collusion-resistance: An adversary that holds multiple keys should only be able to access data if at least one individual key grants access.
Attribute-based encryption (ABE) is a relatively recent approach that reconsiders the concept of public-key cryptography. In traditional public-key cryptography, a message is encrypted for a specific receiver using the receiver's public-key. Identity-based cryptography and in particular identity-based encryption (IBE) changed the traditional understanding of public-key cryptography by allowing the public-key to be an arbitrary string, e.g., the email address of the receiver. ABE goes one step further and defines the identity not atomic but as a set of attributes, e.g., roles, and messages can be encrypted with respect to subsets of attributes (key-policy ABE - KP-ABE) or policies defined over a set of attributes (ciphertext-policy ABE - CP-ABE). The key issue is, that someone should only be able to decrypt a ciphertext if the person holds a key for "matching attributes" (more below) where user keys are always issued by some trusted party.

## VIII.CONCLUSION

Based on our review of literature on deniable encryption schemes, the user can, in past schemes, come up with only one fake message as a counterpart to defend its secret. In contrast, our scheme allows us to bring more than one fake message to hide a secret. Also, this encryption scheme is editable in the sense that at any instant in time, we can update our message by changing only one element in the cipher text or by changing the key indices. Furthermore, the scheme gives us the freedom to delete any message encrypted inside the cipher text simply by replacing at least one element from the cipher text with a random number. Deleting a message gives us the flexibility to prevent the user who was earlier accessing that message from doing it again. If the user wants to decrypt the cipher text with the older key, the outcome will undoubtedly dissatisfy him. Consequently, False-Bottom Encryption extends deniable encryption by the functionality of adding, editing and deleting possibly several plaintexts inside the cipher text. Our security definition does not account for adversaries profiling the access patterns of a user, which calls for additional techniques to either randomize or ‘‘equalize’’ all access sequences.

Future work will thus investigate extensions to our scheme by means of private information retrieval or other techniques (see the related work, in particular), to analyze if information-theoretic security remains accomplishable or deteriorates against attackers that profile the (physical) device usage.

## REFERENCES

[1] R. Canetti, U. Feige, O. Goldreich, and M. Naor, ‘‘Adaptively secure multi-party computation,’’ in Proc. 28th Annu. ACM Symp. Theory Com- put., Philadelphia, PA, USA, 1996, pp. 639–648. [Online]. Available: http://portal.acm.org/citation.cfm?doid=237814.238015.

[2] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, ‘‘Deniable encryption,’’ in Proc. 17th Annu. Int. Cryptol. Conf. (Lecture Notes in Computer Sci- ence), vol. 1294. Santa Barbara, CA, USA: Springer, 1997, pp. 90–104.

[3] A. Juels and T. Ristenpart, ‘‘Honey encryption: Security beyond the brute-force bound,’’ in Advances in Cryptology—EUROCRYPT (Lec- ture Notes in Computer Science), vol. 8441, D. Hutchison, T. Kanade.

[4] J. Kittler, J. M. Kleinberg, A. Kobsa, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, D. Terzopoulos, D. Tygar, G. Weikum, P. Q. Nguyen, and E. Oswald, Eds. Berlin, Germany: Springer, 2014, pp. 293–310, doi: 10.1007/978-3-642-55220-5_17.

[5] Ravindra Changala, "Sentiment Analysis in Social Media Using Deep Learning Techniques", International Journal of Intelligent Systems and Applications In Engineering, 2024, 12(3), 1588–1597.

[6] Ravindra Changala, “Integration of IoT and DNN Model to Support the Precision Crop”, International Journal of Intelligent Systems and Applications in Engineering, Volume 12, Issue 16s), February 2024.

[7] A. O'Neill, C. Peikert, and B. Waters, ''Bi-deniable public-key encryption,'' in Advances in Cryptology—CRYPTO (Lecture Notes in Computer Science), vol. 6841, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg.

[8] F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. P. Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, and P. Rogaway, Eds. Berlin, Germany: Springer, 2011, pp. 525–542, doi: 10.1007/978-3-642-22792-9_30.

[9] M. Klonowski, P. Kubiak, and M. Kutylowski, ''Practical deniable encryp-tion,'' in SOFSEM 2008: Theory and Practice of Computer Science (Lecture Notes in Computer Science), V. Geffert, J. Karhumäki, A. Bertoni,B. Preneel, P. Návrat, and M. Bieliková, Eds. Berlin, Germany: Springer, 2008, pp. 599–609.

[10] Ravindra Changala, "UI/UX Design for Online Learning approach by Predictive Student Experience", 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA 2023), DVD Part Number: CFP23J88-DVD; ISBN: 979-8-3503-4059-4.

[11] Ravindra Changala, "Optimization of Irrigation and Herbicides Using Artificial Intelligence in Agriculture", International Journal of Intelligent Systems and Applications in Engineering, Volume 11, Issue 3), July 2023.

[12] Ravindra Changala, "Evaluation and Analysis of Discovered Patterns Using Pattern Classification Methods in Text Mining" in ARPN Journal of Engineering and Applied Sciences, Volume 13, Issue 11, Pages 3706-3717 with ISSN:1819-6608 in June 2018.

[13] R. Canetti, S. Park, and O. Poburinnaya, ''Fully deniable interactive encryption,'' in Advances in Cryptology—CRYPTO (Lecture Notes in Computer Science), vol. 12170, D. Micciancio and T. Ristenpart, Eds. Cham, Switzerland: Springer, 2020, pp. 807–835, doi: 10.1007/978-3-030-56784-2_27.

[14] C. Dwork, M. Naor, and A. Sahai, ''Concurrent zero-knowledge,'' J. ACM, vol. 51, no. 6, p. 851–898, Nov. 2004, doi: 10.1145/1039488.1039489.

[15] M. Naor, ''Deniable ring authentication,'' in Advances in Cryptology— CRYPTO (Lecture Notes in Computer Science), vol. 2442, G. Goos.

[16] J. Hartmanis, J. van Leeuwen, and M. Yung, Eds. Berlin, Germany: Springer, 2002, pp. 481–498, doi: 10.1007/3-540-45708-9_31.

[17] Ravindra Changala, Framework for Virtualized Network Functions (VNFs) in Cloud of Things Based on Network Traffic Services, International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169 Volume 11, Issue 11s, August 2023.

[18] Ravindra Changala, Block Chain and Machine Learning Models to Evaluate Faults in the Smart Manufacturing System, International Journal of Scientific Research in Science and Technology, Volume 10, Issue 5, ISSN: 2395-6011, Page Number 247-255, September-October-2023.

[19] Ravindra Changala, A Novel Prediction Model to Analyze Evolutionary Trends and Patterns in Forecasting of Crime Data Using Data Mining and Big Data Analytics, Mukt Shabd Journal, Volume XI, Issue X, October 2022, ISSN NO: 2347-3150.

[20] J. Li, Q. Yu, and Y. Zhang, ''Hierarchical attribute basedencryption with continuous leakage-resilience,'' Inf. Sci., vol. 484, pp. 113–134, May 2019. [Online]. Available: https://linkinghub.elsevier. com/retrieve/pii/S0020025519300684.

[21] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, ''Flexible and fine-grained attribute-based data storage in cloud computing,'' IEEE Trans. Services Comput., vol. 10, no. 5, pp. 785–796, Sep. 2017.