International Journal of Engineering Research & Management Technology
ISSN: 2348-4039
March-April-2019 Volume 6, Issue-2
Email: editor@ijermt.org www.ijermt.org

# SECURING DATA DURING TRANSMISSION AND STORAGE USING SNAD ENCRYPTION METHOD

**Dr. Vikas Jain,**

Assistant Professor, SCRIET-DCA,

Ch. Charan Singh University, Meerut, Uttar Pradesh, India

**Abstract:**

*In this day and age, it is of the utmost importance to guarantee the safety of data while it is being sent and stored in order to safeguard sensitive information from being accessed by unauthorized parties and attacks from cybercriminals. The purpose of this study is to investigate the many methods that may be utilized to protect data, with a particular focus on encryption strategies, secure communication protocols, and resilient storage options. It is necessary to employ end-to-end encryption and secure socket layer (SSL) protocols in order to establish a secure communication channel at the time of transmission since data is susceptible to being intercepted and altered while it is being transmitted. Encryption techniques, such as the Advanced Encryption Standard (AES), and the installation of secure key management practices are essential for preventing unwanted access to data that is stored in a safe location and ensuring that the data is not compromised. In addition, this article investigates the role that access control methods, frequent security audits, and compliance with regulatory requirements play in the process of strengthening data security. The combination of these techniques not only reduces the likelihood of adverse outcomes, but it also strengthens the overall resilience of information systems in the face of constantly developing cyber threats. It is possible for enterprises to properly protect their digital assets and continue to keep the trust of their stakeholders if they adopt a holistic strategy to data security.*

***Keywords:*** *Securing, Data Transmission, Storage, Encryption*

## Introduction:

The growth of digital data in today's linked world has completely altered the ways in which corporations execute their operations, interact with one another, and store information. Nevertheless, this digital transition is accompanied with substantial hazards, particularly with regard to the protection of data while it is being sent and after it has been stored. It is becoming increasingly usual for there to be breaches in cybersecurity, theft of data, and illegal access, which highlights the crucial necessity for effective security measures. It is impossible to exaggerate the significance of data security, since it is an essential component in ensuring the availability, integrity, and confidentiality of sensitive information. The transmission of data involves the movement of information over networks, which makes it vulnerable to assaults such as eavesdropping, interception, and man-in-the-middle attacks. In a similar vein, data that is kept on various media and devices is susceptible to several types of risks, including physical theft, illegal access, and dangerous software. As a result, the protection of data during these stages calls for an all-encompassing strategy that incorporates sophisticated encryption methods, secure communication protocols, and tight access control measures. Encryption is a fundamental component of data security because it converts data that can be read into a format that cannot be read and can only be decoded by individuals with the

appropriate authorization. The safe Socket Layer (SSL) and Transport Layer Security (TLS) protocols are responsible for ensuring that communication channels are safe. These protocols prevent data from being intercepted and tampered with while it is being transmitted. A further point to consider is the importance of implementing secure key management processes in order to prevent unwanted access to encryption keys. When it comes to data that is not actively being used, it is essential to make use of robust encryption techniques such as the Advanced Encryption Standard (AES). In addition to this, the implementation of access control mechanisms, the performance of routine security audits, and the observance of regulatory standards work together to strengthen data storage against potential dangers. The confluence of these tactics results in the creation of a robust security architecture that not only safeguards data but also improves the trust and compliance of the company. For the purpose of this study, we will investigate the various methods and best practices that are available for safeguarding data while it is being sent and stored. By gaining a knowledge of these steps and putting them into practice, companies may considerably reduce risks and protect their digital assets in a cyber world that is becoming increasingly hostile. Not only is the protection of data a difficult technological task, but it is also a crucial legislative obligation in many different businesses. It is necessary to take severe precautions in order to secure personal and sensitive information in order to comply with legislation such as the General Data Protection Regulation (GDPR) in Europe, the Health Insurance Portability and Accountability Act (HIPAA) in the United States, and other data protection standards throughout the world. The failure of an organization to comply with these requirements may result in significant penalties, legal implications, and harm to the company's image. Keeping up with the ever-changing threat landscape makes the mission of data protection even more difficult. Because cybercriminals are always developing more complex methods to exploit weaknesses in systems, it is necessary to have a proactive and dynamic approach to security. Among these are the utilization of cutting-edge security technology, the cultivation of a culture of security awareness inside enterprises, and the maintenance of a current understanding of the most recent dangers. Additionally, the proliferation of cloud computing and the Internet of Things (IoT) adds new layers of complexity to the complexity landscape. Data today often moves across a variety of settings that are deployed across several locations, necessitating a comprehensive security plan that incorporates cloud infrastructure, endpoint devices, and traditional information technology systems. When it comes to ensuring the validity and integrity of data, secure data transfer in these kinds of situations frequently depends on a mix of Virtual Private Networks (VPNs), secure application programming interfaces (APIs), and blockchain technology. The purpose of this article is to offer a detailed review of the protocols and technologies that are utilized to ensure the safety of data while it is being sent and stored. Methods of encryption, secure protocols, key management, and the most effective procedures for access control will all be covered in this presentation. In addition to this, it will investigate new technologies and trends that have the potential to improve data security in the years to come. Stakeholders, including information technology workers, security specialists, and organizational leaders, may better secure their data assets against unauthorized access and cyber threats if they have a solid knowledge of the concepts and practices that are presented in this paper. This expertise is essential for ensuring that contemporary companies continue to be successful and resilient, as well as for preserving the confidence of their customers, partners, and regulatory agencies.

## CRITERIA FOR EVALUATION

The purpose of this section is to develop a standard set of criteria that can be used to evaluate a storage security solution. It is possible to approach storage systems in a variety of different ways; nevertheless, for

the goal of providing a common reference, the following characteristics have been chosen: secrecy, integrity, availability, and performance. It is vital to define the assessment criteria before evaluating the particular systems, despite the fact that this study does not approach any of the criteria in a thorough manner. Within the realm of computer security, the terms confidentiality, integrity, and availability are frequently used. Performance was also included in order to guarantee that systems strike an adequate balance between the capacity to handle information and the level of security they provide. First things first: before we get into more depth about each component, it is essential to realize that none of these characteristics are incompatible with one another. In fact, in order to have a safe system, each and every one of these characteristics must be met.

**Confidentiality**

In the context of information security, protecting confidentiality means ensuring that no one has access to the data unless they have been officially permitted to do so. These permission procedures are controlled in a variety of different ways by distinct systems. In order to correctly identify people through authentication, the first stage in the process of granting access to information is to do so. A user must be correctly identified before getting access to the storage system, and after that person has been adequately recognized, the system must only provide access to the data that is connected with that user. The storage system must describe the ways by which a user can be properly identified. Having the appropriate authority to access the storage system does not always mean that one has access to the entire system. In reality, the concept of least privilege is typically being used in most situations. Those who control the data, on the other hand, are required to put a system in place that allows them to delegate authority to other people so that they can access information when it is necessary. Not only does maintaining confidentiality need the system to manage authorization to data, but it also necessitates the encryption of data in order to avoid information assaults. Consequently, the system must need the application of cryptographic keys by either the computers or the servers. There has been a considerable influence on the overall storage approach as a result of the different design decisions that were made between user controlled keys and server managed keys. In order for numerous users to be able to share information, it is necessary for them to have access to the proper keys. Whether the keys are distributed by a centralized group server or by individual file owners, it is necessary to examine the impact on both performance and user comfort. In this work, the study of key management is not meant to provide a detailed explanation of cryptography; rather, it is vital to have a grasp of how keys are distributed and used in order to have a complete comprehension of the broader system. Therefore, it is necessary to have a grasp of how a specific system handles keys. This is because the cryptographic procedures are frequently the most computationally costly portion of retrieving data that has been safely stored. The manner in which keys are revoked is related to key management and is an additional important topic of debate. As soon as an owner or administrator makes the decision to revoke a specific user's access to data, the keys that the user possessed must either no longer provide access to the system or, if they do permit access, they must not permit access to any future versions of the files. The work that is necessary to re-encrypt data in order to maintain secrecy is the manifestation of the cost that is associated with canceling a user profile. Due to the fact that duplicates might have been generated, it is not feasible to physically revoke a user's keys in order to prevent that user from being able to execute actions. Therefore, the system must render all of the keys of a revoked user outdated and re-encrypt all of the data with a new key. After then, the issue that has arisen revolves once more on the compromise that must be made between performance and security. Lazy revocation and aggressive revocation are the two

basic approaches that may be utilized to secure the data once the key has been revoked. When lazy revocation is utilized, the system does not re-encrypt the data that the revoked user had previously been authorized to access until the subsequent valid user makes an attempt to access the file. The expense is effectively defrayed over time by this method; but, the data will remain susceptible to the user whose access has been terminated for an undetermined amount of time. The aggressive revocation method, on the other hand, promptly re-encrypts all of the files that the user whose access has been revoked may potentially access. Once the encryption has been re-encrypted, new keys will need to be issued to all of the persons who are affected by the new encryption. This will add more weight to the key distribution method, and it is obvious that this process will take some time. On the other hand, aggressive revocation makes a sacrifice of time in order to increase security, whereas lazy re-encryption makes a compromise of some security in order to save time.

**Integrity**

The concept of integrity encompasses a wide range of subject matter, one of which is the maintenance of data consistency in the face of both unintentional and purposeful attacks on data. In the context of this study, the scope of the integrity analysis is restricted to the procedures that are utilized to prevent the intentional change or destruction of information. The assumption that arises as a consequence of this is that when a user accesses information that has been saved, absolutely no data has been altered without authorization. The integrity of many systems is maintained by verifying that the data originates from the location that was anticipated. When it comes to data that has been saved, the concept of integrity makes it clear that the files on the disk have not been altered. The processes that are used to ensure integrity may be divided into two categories: those that prohibit data alteration and those that detect information modification. Along the same lines as confidentiality, modification prevention mandates that users must first get authorization before making any changes to files and dictates that files can only be altered in a manner that has been authorized. The difference between integrity and confidentiality lies in the fact that the former is concerned only with determining whether or not the data has been compromised, whilst the latter is concerned with ensuring that the data is accurate. It is generally accepted that detection techniques operate on the assumption that assaults are unavoidable and that there must be appropriate methods to evaluate any harm that has been done, recover from the damage, and apply the lessons gained to future preventative mechanisms.

**Availability**

The amount of time, space, and representation that is available is taken into consideration in this study. The information must be accessible to a user who is permitted to access it within a reasonable amount of time, without using up all of the storage space that is available, and in a format that is easily understood. In order to prevent an adversary from preventing authorized access to information through a denial of service attack, a system cannot allow this to happen. Both the aims of availability and those of secrecy must be taken into consideration within the realm of security. It is essential to keep in mind that the two goals are in some degree in conflict with one another.

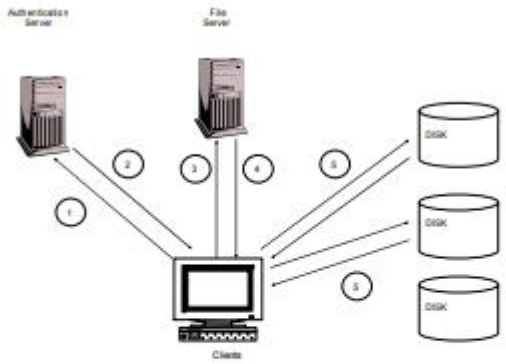
**Performance**

Conflicts can arise between the level of security and the performance of the system. The system's performance is negatively impacted as a result of the necessity of providing the necessary layers of security

in order to prevent damaging assaults. Both the objective of an efficient system and the goal of a safe environment are inherently in tension with one another. Every extra security measure necessitates processing that is prohibitively expensive in terms of computing, which in turn hinders the system's capacity to carry out other activities; all security measures constitute an additional burden for the system. The performance cost that is linked with the specific measurements of the system is something that each of the storage approaches that were studied tries to mitigate as much as possible. Due to the fact that encryption is computationally costly, it is linked with the most significant performance penalty. Both the encrypt-on-wire and the encrypt-on-disk methods to storage security are fundamentally distinct from one another. As a result, the responsibility of encryption is distributed among several components of the system. The authors Riedel et al. offer a comprehensive analysis of the comparisons and contrasts between the two.

**Secure Disks for Network Attached Devices (NASD)**

When a client wants to access data in a standard distributed file system, they are required to send a request to the file server. After then, the server is obligated to validate the authorization of the client and will only release the file if the necessary conditions are satisfied. The server has the potential to become a bottleneck very rapidly due to the fact that it must respond to each and every file access request made by each and every client. Through a single interaction with a user and the provision of a "capability key," the primary objective of NASD is to alleviate the bottleneck that is caused by the server. Using the capability key, the user is able to immediately access the disk(s) that are relevant to their needs without any additional interaction with the server. It is necessary for the disks themselves to be "intelligent" in the sense that they must have sufficient internal capabilities to process the capability key and directly manage file access requests..



1. Client requests authentication
2. Server responds
3. Client request to file server
4. File server responds with capability object
5. Request directly to drive
6. Response from disk

**Figure 1 NASD**

Confidentiality. In the architecture of the NASD, there are two servers: the first server is responsible for authentication, and the second server is the real file server. Although the NASD does not define the authentication mechanism, it does propose that any current approach that is comparable to Kerberos be utilized. Immediately after receiving authentication, a user will submit a request to the server that stores

files. Following the completion of the authentication process, the server will then supply the user with a capability key that matches to the user's permissions to access the file in question. Once the capability key has been obtained, a user is able to interact directly with the data disk for any and all future access requests that may be made during a particular session. When it comes to the system's integrity and secrecy, the capability object is the most important factor to consider. When a file manager grants a client's request for access, the file manager delivers the client a capability token and a capability key in a private message. These two pieces of information come together to form a capability object. The key is a message authentication code (MAC) that is comprised of the capabilities and a secret key that is exchanged between the file server and the physical disk drive. The token is the one that comprises the access privileges that are being given for the request. After then, clients are able to send a direct request to a NASD drive using the capability object that they have provided. The capability token is then interpreted by the drive, which utilizes the secret key that it shares with the file server. This allows the drive to validate the user's access privileges and fulfill the request it has received. Due to the fact that the MAC can only be understood by utilizing the shared secret key between the disk and the server, any changes made to the arguments or submission of incorrect arguments will result in the request being rejected.

Integrity. The innovative idea that is linked with NASD is the utilization of the disks themselves to fulfill a portion of the data integrity requirement. In addition to encrypting data and transmitting results to clients, the "intelligent" disks are responsible for interpreting the capability objects. The disk employs the same hash MAC combination that it used to enable a client access in order to encrypt and transfer the data to the client. This process is done in order to guarantee the integrity of the data on the client end. Following this, the client is able to check the authenticity of the transmission while the decryption process is taking place.

Availability. The scalability of the system is improved by the fact that NASD permits direct access to the disk. The throughput of the system grows linearly with the number of clients and disks connected to it. On the other hand, because the file server must be trusted in order to first supply capability keys, the server itself constitutes a single point of attack. There is no mechanism to take preventative measures against a denial of service attack in the event that the server is hacked.

Performance. The ability to expand bandwidth linearly with the number of disks in the system is one of the driving factors for adopting NASD. However, the expense of cryptography somewhat offsets these gains, thus it is important to keep this in mind. The twofold cost of cryptographic operations that are incurred as a result of the encrypt-on-wire approach is a significant and significant performance issue that is linked with NASD. It is necessary to encrypt every data transfer before it is delivered, and then decrypt it once it reaches its destination, whether it is from the disk to the client or from the client to the disk. The National Advanced Security Data (NASD) utilizes a cryptographic method known as "hash and MAC" rather than a normal MAC in an effort to lessen the performance cost. Within the context of a conventional MAC method, the secret key of a client is utilized throughout the calculation. The hash and MAC algorithms, on the other hand, make use of the raw data from the file in order to precompute a sequence of message digests that are generic for the file in question. Only when a client makes a request for a file does Hash and MAC perform the application of a client's secret key to the message digests. As a consequence of this, the secret key is only required for a tiny portion of the total calculation, which results in a considerable reduction in the latency that is associated with on-the-fly cryptography. It was established through experiments that the delay for utilizing cryptographic procedures was constrained by a twenty percent increase in the amount of time required to serve a request in comparison to a request that did not use cryptography.

**PASIS - Storage for the Long Term**

The PASIS system is a survivable storage system that was developed to solve the issues that are connected with hacked servers. The system operates under the assumption that compromised servers will exist and, as a result, tackles the issue of how to safeguard data in an environment like this. PASIS makes use of a threshold approach to distribute trust across storage nodes in order to prevent data security breaches even when confronted with a server that has had unauthorized access. The threshold system is responsible for encoding, replicating, and dividing information in such a way that the individual parts of data are kept in distinct locations. An "agent" on the client side is required by PASIS in order to comprehend user-level requests and the answers that are associated with them from the numerous PASIS servers that are linked to the storage nodes. This is done in order to make the data dispersal visible to the users.

Confidentiality. The PASIS system works to prevent the compromise of data by keeping components of a file in several locations. This ensures that a single server that has been compromised cannot provide any information that is pertinent to the situation. PASIS employs a ramp p-m-n threshold approach to provide secrecy rather than encryption. This scheme splits data into n shares in such a way that any m of the shares may reconstruct the original data, but revealing information about the original data is impossible for less than p shares. (The level of security can be increased by combining cryptography with a threshold mechanism; however, any encryption would need to be put on top of PASIS in order to do this). There will be no information that is compromised so long as there are less than p shares that are ever exposed to an intruder.

Integrity. PASIS is able to contribute to the maintenance of data integrity since it does not rely on any particular collection of PASIS servers to supply the necessary m shares of the data. Due to the fact that the client agent is able to rebuild the original data when they possess any set of m shares, those m shares might originate from any of the numerous servers that are interconnected in the network. It is necessary for an intruder to hack the m servers that are fulfilling the request and modify the data in order to prevent the integrity of the data. The request is rebroadcast in the event that the client agent does not get the required number of shares or is unable to rebuild the original file as a result of malicious activity.

Availability. In a manner that is analogous to the argument that was presented in support of the PASIS integrity upgrades, the fact that the system requires just m shares in order to retrieve data in order to boost data availability in the event that servers fail. There is an upper constraint of m for the number of servers that are required in the "surviving" subset of servers. This means that even if (n-m) servers are hacked or unavailable, the system will still be able to properly handle the request. Likewise, when a write operation is performed, the system administrator has the ability to ascertain the number of shares that are necessary to accept the write. It is necessary to correctly write at least m shares; however, any number between n and m will yield accurate data. It should come as no surprise that the shares that have been written with better success will give greater availability for subsequent file access requests. PASIS does not directly handle concurrent access or concurrent edits to files, despite the fact that it makes it possible for single users to have access to more data than before. This indicates that an extra method must be put on top of PASIS in order to guarantee atomicity. This, in turn, indicates that there may be a chance for message forwarding overhead or latency when many users access the same file at the same time.

Performance. When compared to a conventional distributed file system, PASIS is hampered by the fact that it requires a greater number of message passes in order to obtain the same information. When compared to PASIS, which must broadcast requests to at least m servers and then aggregate the messages that are generated on the client computer, a typical system only makes a request to a single server. Due to the fact that there are significant performance trade-offs involved with picking alternative values for n-m-p, it is difficult to estimate the overhead that is associated with PASIS. Nevertheless, the values can be altered to suit the requirements of a specific file. An example of this would be raising the value of n, which would raise the possibility that m shares will be available; but, this would also imply that more shares of the file would be kept, which would increase the likelihood that theft would occur. Users have the ability to pick acceptable values for each file that they save in the system, which is an advantage that comes with this freedom. During the course of their investigation, the designers of PASIS found that accessing tiny files resulted in a high performance cost, but accessing big files resulted in a penalty that was minimal.

**"SNAD" stands for "secure network attached disks."**

By encrypting all of the data and allowing decryption to take place only on a client system, Secure Network Attached Disks are meant to prevent any unauthorized personnel from accessing the contents that are securely stored on the disk. This eliminates the possible harm that may have been presented by physically seizing a disk or by compromising the access permissions of the system administrator. In order to decrypt any data on their own, the individual drives do not possess sufficient information. Instead, they rely on a key management mechanism that supplies an authorized user with necessary keys to decrypt files on the distant client workstation. Observance of strict confidentiality. It is the lockbox mechanism that is responsible for storing keys that is the most important functionality underpinning SNAD. separately encrypted with a symmetric object key, each file is made up of secure data objects of varying sizes. These data objects are all encrypted separately. There is a pointer to a key object for that file that is contained within the metadata of the file. This key object can be regarded a file in and of itself. A unique key file ID, the user ID of the person who created the file, and a digital signature from the person who modified the file most recently are all fields that may be found inside the metadata of the key object. All other users are supplied with the digital signature of the person who made the most recent modification to the file in order to provide them with the assurance that the key object itself has not been altered. Any authorized user may check that the signature of the key object corresponds with someone who is specifically permitted to write to the file. The key object file itself is made up of tuples that correspond to valid users for the key object file that was originally created. A user ID field, the object symmetric key that is used to access the secure data objects, and a listing of whether or not the user has permission to write to the key object (which coincides with permission to write to the original file) are all included in each tuple. Because the object key is encrypted with the public component of a user's asymmetric key, the only method to decode the object key is with the user's private key on the user's client computer. This is because the object key is encrypted with the public portion of Asymmetric Key. Because of this, it is impossible for any unauthorized individual to ever have access to the symmetric key that was used to encrypt the data that was saved. Additionally, SNAD manages authorized users by maintaining a certificate object that contains tuples that contain valid user IDs, the user's public key, a hashed message authentication code key to provide and verify user digital signatures, and a timestamp that the system updates whenever the user performs a write operation in order to prevent replay attacks. This is done in order to prevent replay attacks.

Integrity. For the purpose of providing an upgrade to the integrity of the data, SNAD keeps a non-linear checksum of the original data alongside the encrypted data. This allows the user to verify that the file has not been altered maliciously while it was being stored within the system. When an authorized user makes a change to the file, the checksum is updated to reflect the new version of the file. Examining the information of the key object file and confirming the digital signature that is supplied are two more methods that users may use to validate the authenticity of writes.

Availability. Users must have access to the lock-box of keys in order to access files. This is because the lock-box of keys is an essential component of SNAD. Unfortunately, the lockbox is only stored on a single trusted server, which puts an opponent in a position where they have only one port of entry to attack. There is no way to ensure that a denial of service attack will not occur in the case that the lockbox server is hacked. In addition, the system does not have a key revocation policy that is established, and the decision about whether to implement active or lazy revocation is left up to the owner of the file.

Performance. The client computers are responsible for doing the computationally intensive encryption and decryption activities. This helps to eliminate any possible bottlenecks that may occur at the server. The process of decryption, on the other hand, might still be exceedingly sluggish even when it is carried out on client devices that are comparably quicker. In order to provide users with a variety of alternatives, the developers of SNAD have created three distinct techniques for the provision of digital signatures. These schemes compromise security at the expense of performance. The creator of a file has the ability to choose the level of granularity that will be used to verify the digital signature. The higher the level of granularity, the greater the level of security that is supplied, and vice versa. Through the use of empirical investigations, the developers have demonstrated that the procedure of digital signature is by far the most expensive with SNAD For the most part, the most secure solution, which involves providing digital signatures with each block write and verifying them with each block read, is not acceptable for usage in normal situations. Only the option that was the least secure of the three—verifying the digital signature based on a hashed MAC rather than a public key—was found to be equal to a system that did not have any security.

**Conclusion:**

A key part of contemporary cybersecurity measures is protecting data while it is in transit and storage. Organizations need to take strong and thorough precautions to safeguard sensitive data as the amount of digital data keeps increasing and cyber threats change. Organizations may greatly lessen the likelihood of data breaches and illegal access by utilizing sophisticated encryption methods, secure communication protocols, and rigorous access control systems. The transfer of sensitive information is protected from eavesdropping and manipulation by using end-to-end encryption and SSL protocols. Secure key management techniques and robust encryption algorithms, such as Advanced Encryption Standard (AES), are crucial for protecting data when it is at rest from prying eyes. To further strengthen data storage against any threats, it is essential to implement access control mechanisms, conduct frequent security audits, and ensure compliance with regulatory standards. A proactive and adaptable strategy is required for data protection due to the ever-changing nature of cyber threats. In order to implement a successful cybersecurity plan, firms must stay informed about emerging threats, use state-of-the-art security solutions, and promote a security-aware culture. The growing importance of cloud computing and the IoT highlights the necessity for a comprehensive security strategy that covers many dispersed contexts. Businesses may strengthen their defenses against cyberattacks and safeguard their data by studying up on the best practices

and approaches outlined in this article. Technical requirements, legal requirements, and stakeholder confidence all hinge on data security measures that guarantee data is accessible, intact, and secret at all times. Ultimately, in the modern digital world, it is crucial to have a strong data security plan that covers storage as well as transmission. Companies that put data security first will have a greater chance of surviving the ever-changing cyber threat landscape, meeting regulatory standards, and maintaining operational and reputational integrity.

## REFERENCES

[1] Mehmet Bakkaloglu, Jay J. Wylie, Chenxi Wang, Gregory R. Ganger. On Correlated Failures in Survivable Storage Systems. CMU SCS Technical Report CMU-CS- 02-129. May 2002.

[2] Scott A. Banachowski, Zachary N. J. Peterson, Ethan L. Miller, and Scott A. Brandt. Intra-file security for a distributed file system. In Proceedings of the 19th IEEE Symposium on Mass Storage Systems and Technologies, pages 153–163, College Park, MD, April 2002. IEEE.

[3] Matt Blaze, A Cryptographic File System for Unix, First ACM Conference on Communications and Computing Security, Fairfax, VA November, 1993.

[4] Brian Cornell, Peter A. Dinda, Fabian E. Bustamante Wayback: A User-level Versioning File System for Linux. In Usenix Annual Technical Conference, Boston, MA Jun 27 – Jul 2, 2004

[5] William Freeman and Ethan Miller. Design for a decentralized security system for network-attached storage. In Proceedings of the 17th IEEE Symposium on Mass Storage Systems and Technologies, pages 361–373, College Park, MD, March 2000.

[6] Kevin E. Fu. Group Sharing and Random Access in Cryptographic Storage File Systems. Massachusetts Institute of Technology, Jun 1999. (Cepheus)

[7] Kevin Fu, M. Frans Kaashoek, David Mazieres. Fast and Secure Distributed Read Only File System. In Proceedings of the 4th USENIX Symposium on Operating Systems Design and Implementation, pages 181-196, Sand Diego, CA, Oct 2000.

[8] Gregory R. Ganger, Pradeep K. Khosla, Mehmet Bakkaloglu, Michael W. Bigrigg, Garth R. Goodson, Semih Oguz, Vijay Pandurangan, Craig A. N. Soules, John D. Strunk, Jay J. Wylie. Survivable Storage Systems. DARPA Information Survivability Conference and Exposition (Anaheim, CA, 12-14 June 2001), pages 184-195 vol 2. IEEE, 2001.

[9] Garth Gibson, David Nagle, Khalil Amiri, Fay Chang, Howard Gobioff, Erik Riedel, David Rochberg, Jim Zelenka, "Filesystems for Network-Attached Secure Disks" CMU Computer Science Technical Report, CMU-CS-97-118. July 1997.

[10] Howard Gobioff, David Nagle, Garth Gibson, “Embedded Security for NetworkAttached Storage”, CMU SCS technical report CMU-CS-99-154, June 1999.