Decentralized Key Management:

Architecture, Threats, and Applications.

Navya Shah

USC ID: 4062254582

CSCI 530

I have read the Guide to Avoiding Plagiarism published by the student affairs office. I understand what is expected of me with respect to properly citing sources, and how to avoid representing the work of others as my own. The material in this paper was written by me, except for such material that is quoted or indented and properly cited to indicated the sources of the material. I understand that using the words of others, and simply tagging the sentence, paragraph, or section with a tag to the copied source does not constitute proper citation and that if such material is used verbatim or paraphrased it must be specifically conveyed (such as through the use of quotation marks or indentation) together with the citation. I further understand that overuse of properly cited quotations to avoid conveying the information in my own words, while it will not subject me to disciplinary action, does convey to the instructor that I do not understand the material enough to explain it in my own words, and will likely result in a lesser grade on the paper.

Signed: Navya Shah

Introduction:

Key management sits at the core of every system. It determines who can access the data, how trust is formed, and how identities are verified. Key management services are divided into two main models: centralized and distributed. Centralized key management is where the service is controlled by a single entity in authority. For distributed, the responsibilities are shared among multiple entities within the network. However, in reality, distributed key management only works in theory as it requires a centralized authority to carry out key distribution. This moves us away from traditional methods and leads us to explore new developments in technology, such as utilizing a blockchain-based distributed key management solution. This would not only reduce the risks of centralized key management but also decentralize the storage of keys across multiple nodes of the blockchain. This paper examines how this decentralized model operates, its differences from PKI, and the security principles it preserves or challenges. Through analysis of identity frameworks, key protocols, real world applications, and a case study of a decentralized key management system, the goal is to understand how trust is engineered in systems that work to remove the support of traditional authority structures.

Traditional vs Decentralized Key Management

Regardless of the key management service being centralized, distributed, or decentralized, it all begins at the root of trust, which is the starting point of the chain of trust from which nothing more can be derived. It serves to be an anchor, and it takes on a different form for each model of key management (Preukschat and Reed). Centralized PKI relies on hierarchical trust, which makes the certificate authorities act as that trust anchor for all issued keys and credentials. Here, the Certificate Authority (CA) would verify a client's identity, sign the public keys, and issue

certificates so that those identities are bound to those keys. This puts the trust component on the CA who has verified the identity in the form of a certificate. The root CA, which would be a part of a small number of highly privileged authorities, is where the trust comes from, as there can be intermediate CAs involved, but they would point back to the root (Preukschat and Reed). PKI depends on mechanisms such as certificate revocation lists (CRLs) or online status checking (OCSP) to be alerted when a key is no longer valid. However, the effectiveness of these mechanisms would depend on the security and availability of the CA. This points out how PKI puts a great deal of trust in a small part of the system, which is the CA, as it would be an essential component of the Trusted Computing Base (TCB). This means that if CA is trustworthy, then the system functions smoothly. However, in the event of compromise, the trust foundation collapses (Preukschat and Reed).

When considering compromise, it's important to know that PKI possesses certain structural weaknesses that grow to become more severe as systems scale. Having a single point of failure is a significant challenge as having a compromised CA would lead to the issuance of fraudulent certificates, which would allow attackers to impersonate legitimate users (Ni et al.). There is also the problem of revocation as CRLs can grow large and propagate slowly. OSCP responses would also rely on the availability of CA infrastructure to be able to check the validity of the certificates (Ni et al.). PKI additionally struggles with having efficient key rotation and communication across domains where different organizations must trust each other's CA to exchange keys. This inefficiency is due to excessive reliance on centralization, which makes PKI "unsuitable for dynamic distributed network environments," (Ni et al.). There is also a large dependence on having proper configurations, as incorrect settings or having expired roots can block access or open security gaps. Centralized key management tries to address this by having

multiple nodes, which creates distributed key management services. However, traditional distributed management would ultimately rely on centralized entities to distribute keys, which increases security vulnerabilities and trust issues. These problems become especially paramount when working with groups with many members or frequent changes (Ni et al.). This makes centralized approaches, whether working in a traditional PKI sense or the distributed way, not very well suited for dynamic distributed environments. This can be particularly noted for IoT or larger scale identity systems, where you have keys that must adaptly quickly and securely (Ni et al.).

IoT networks are embedded in our everyday lives, and we interact with decentralized identity frameworks and work with cross-organization data systems. These are a few examples that use decentralized systems. The actions would require speed and efficiency to provide continuous and trustworthy key operations without relying on a central authority. With evolving technology, it can be seen how traditional centralized or distributed systems end up highlighting PKI's weakness, especially when applying it to a decentralized product, and rather push toward decentralized key management for it as well. When using decentralized key management, it replaces institutional trust with cryptographic trust: identities are verified through proofs of key ownership, and system integrity is maintained through consensus rather than hierarchy (Kim,; Yildiz and Bahtiyar). There are public keys, updates, and revocations that are recorded on a tamper-proof public ledger, which allows every participant to validate its integrity independently. This system distributes CA's responsibility across many nodes, reducing single points of failure and enabling key operations to match the scale and dynamism of modern distributed systems (Dammak et al.; Ni et al.). As a result, decentralized models provide a more suited and ultimately

more resilient architecture for environments that cannot have security depending on a single gatekeeper.

Decentralized Key Management: Core Concepts

At its core, a decentralized key management system begins by eliminating the need for a trusted central authority by instead shifting that responsibility away from a single node toward the network as a whole. By distributing trust across multiple independent participants, this system ends up minimizing the TCB. While it is counterintuitive to assume this, since the CA as a TCB component, comprised of one entity but now would be multiple nodes, the size of the TCB is reduced because no single authority would be completely trusted to behave correctly and that trust would be placed on the network. Decentralized key management moves the concept of correctness coming from the CA to instead enforce correctness through cryptographic mechanisms and distributed validation This design ensures that no single entity must remain uncompromised for the system to function securely, improving resilience against targeted attacks. An IEEE published research paper describes this shift as toward "self-contained trust," where the security of identity and key operations stems from protocol enforcement rather than institutional control (Yildiz and Bahtiyar).

When working with decentralized systems, keys serve as identities, which means that authentication would be achieved by proving the possession of the key rather than the certificate. This structure is formalized by the concept of Decentralized Identifiers (DIDs). A DID is a globally unique identifier that does not depend on any central authority. A DID is generated and controlled by the user instead of being provided by the CA. Each DID corresponds to a DID document, which contains the public keys and authentication methods associated with that

particular identity. This document is stored on a distributed ledger or another tamper-proof system so that anyone can resolve a DID and retrieve the associated public key information and be able to verify signatures from that user (“Decentralized Identifiers”). Identity verification would then be done by checking that the user truly has the private key, which is referenced in the DID. This works because the DID Document is cryptographically bound to the user’s key pair. It can be seen that here identity is no longer verified by an external organization, and it is instead self-sovereign by being cryptographically verifiable, which ultimately makes it independent of centralized trust infrastructures.

Instead of having CA's root of trust, for decentralized systems, we would have a distributed consensus which would record key generation, rotation, and revocation events on a blockchain or distributed ledger to maintain the correct key state. This is done through consensus protocols, which ensure that each update is validated and agreed upon by multiple nodes before it is appended to the ledger. Research on distributed group key management based on the blockchain described how using such a technique would allow all participants to be able to verify key changes independently, which creates a transparent and tamper-proof record of how the keys have changed (Ni et al.). This means that distributed consensus is the mechanisms that enforces the correctness, integrity, and availability of keys and information about identities in the system instead of relying on a certificate authority.

The key lifecycle is different when using decentralized key management systems. Key generation is done locally on devices which avoids risk of generating them on the server. Key distribution would be done by publishing public keys or even encrypted group keys to the ledger which allows all parties to see the same authoritative state. The key rotation is done through changes to the ledger, which is by consensus backed updates that ensure new keys are known and

trusted through all the nodes at the same time (Preukschat and Reed). Revocation here would be done by appending revocation data to the ledger, which moves away from long and inconsistent CRLs. There is also the management of groups in this blockchain-based solution as group protocols are used to maintain secure, dynamic membership in the group by having automatic rekeying when users join or leave (Ni et al.). This does not need a central key server to recalculate and redistribute the keys and it is done much quicker using a protocol (Ni et al.) A significant shortfall of using a decentralized system would be losing the ability to restore a lost private key because there is no authority in place to manage the keys. However, you can preserve access to the systems and recover keys in that way (Preukschat and Reed). To do this, some systems make use of threshold cryptography which is implemented in NuCypher's decentralized key management system. The nodes would collectively perform re-encryption and access control operations without accessing the private keys. This would mean updating the DID to bind it to a new key pair through a ledger-based recovery process, which would be supported by getting multiple parties to authorize it (Yildiz and Bahtiyar). These approaches do not restore the lost key itself but allow users to regain access or rotate to a new key securely, making this key recovery method about secure continuity and not key regeneration.

When comparing decentralized and PKI key management, it is important to highlight the shift in assumptions taking place. Here, trust moves away from institutions and more toward the protocols which is the math behind to provide accuracy and integrity of the data. The user is in so much more control as identity is verified through cryptography of keys, accuracy is upheld by the ledger, and operations are done through updates. This shows how it builds a more resilient foundation for systems that have to operate across large multi-domain environments where centralized trust can be impractical or unsecure.

Identity and Access Without A Central Authority

As established in the previous section, keys serve as identities and decentralized systems extend this concept by building identity and access frameworks that do not depend on a central authority but rather on cryptographic keys to define identity. DIDs were introduced as identifiers that are anchored to a distributed ledger which is essentially the foundation of decentralized identity systems. A DID, being a pointer to a public key, allows users to show they control the private key which validates their identity which allows anyone to verify identity by resolving key information in the ledger and checking signatures ("Decentralized Identifiers").

Self-sovereign identity (SSI) takes this model further by giving users full control over their credentials and how they authenticate. Normally, you would authenticate through centralized identity providers, however using this, users can employ verifiable credentials that are managed locally and show proofs derived from their private keys (Preukschat and Reed). An IEEE published research paper described how this structure would allow authentication to take place entirely using the ledger key bindings and cryptographic proofs which removes the dependency to those identity providers (Yildiz and Bahtiyar).

To compare this system with PKI, decentralized identity differs fundamentally in its approach to validation, revocation and authenticity. PKI puts emphasis on trusting CAs to verify identities, issue certificates and carry out revocations, which all translate into making single points of failure. When using decentralized systems, DIDs are resolved to validate identity by verifying the user has control of that particular private key. Revocation would also be done on the ledger with updates that are immutable and visible to all. An example of this was illustrated in a decentralized healthcare system where patient identities, public keys and access rights were

stored directly on the blockchain rather than being managed by an identity provider. In this model, each patient has an identity on the ledger and healthcare providers would authenticate access requests by checking signatures against the patient's public key. This allows only authorized staff to read and update records. Any changes to the patient's access permission would also be recorded which means unauthorized access is cryptographically impossible without the patient's private key (Deepa et al.). Revocation was also immediate and transparent. However, there is a trade off in usability as users must be responsible for protecting their private keys. In centralized systems, you can always reset a forgotten password or lost credentials through the identity provider. However, now the burden is placed on individuals (Deepa et al.). Alternative recovery processes have methods like social recovery, where there is a threshold of guardians that must approve a recovery event which then updates the DID document and binds it to a new key pair. Another would be threshold-based cryptography where cryptographic secret sharing is used that splits a private key and requires a subset of the splits can reconstruct the key, this allows a threshold of network nodes to authorize re-encryption or policy changes without ever learning the key (Egorov et al.).

Considering authorization when comparing PKI systems to decentralized as well, in PKI systems, authorization would rely on certificates which means the validity of a public key depends on the continued trustworthiness of the issuing institution. In a decentralized ecosystem, this hierarchy is removed as authenticity is established by proving control of the private key linked to a DID document and by relying on the integrity of the ledger to prove the associated public key is genuine. This makes authorization also decentralized as it would refer to identifiers or keys on the ledger to determine if a user should be granted access. Zhang et al./s blockchain-based signature scheme for the smart grid illustrates this foundation by letting any

participant verify the identity of a device directly against ledger keys. While this doesn't define authorization policies, it provides a decentralized authentication layer upon which such policies can be built (Zhang et al.). Similarly, there is also research on decentralized group key protocols that discuss how membership and access rights are enforced through state transitions recorded on the consensus rather than central administrators. In both cases, authorization arises from cryptographic truth recorded on the ledger rather than from certificates issued by a trusted authority.

Example: NuCypher's Decentralized Key Management System

To look at decentralized key management systems more closely, NuCypher KMS serves as an example that uses proxy re-encryption (PRE) to provide cryptographic access control and secure data sharing without using a centralized key server. In this architecture, data owners encrypt the data with their public key and store the ciphertext in the storage layer. Then, Umbral PRE, which is a threshold proxy reencryption scheme, is used for data owners to authorize a recipient, which then generates reencryption keys that allow NuCypher nodes to transform the ciphertext so the recipient can decrypt it using their own private key (Egorov et al.). This process lets a network of intermediate nodes transform the ciphertext so that a different user can decrypt it without the plaintext or private keys being revealed (NuCypher). For access control, the policies and node behavior would be encoded into smart contracts that would outline how reencryption operations are issued and revoked which would inherently replace the central authority of traditional KMS solutions (Egorov et al.). To maintain the authenticity of the policies, nodes can verify the digital signatures of the data owner who created the policies. It can also include reencryption operations,

which can be edited but require consensus from the participating nodes, which prevents unilateral misuse (NuCypher).

Even though NuCypher removes the single point of failure in centralized key servers, its own design introduces problems as well. Since access control relies on a threshold amount of reencryption nodes, it depends on there being enough honest nodes to perform this role. If there are too few nodes participating, then the reencryption cannot take place and authorized users can lose access to data. There is also the limitation of decentralized cryptographic systems that this carries because if a user loses their private key, neither NuCypher or the network can recover it (Egorov et al.). The risk of compromised nodes is prevalent as well since a single node cant successfully act maliciously but if it is a coordinated group of malicious nodes which would be equal to the threshold, then they can work to produce encryption fragments which shouldn't be allowed (Egorov et al.). Lastly, there cannot be corrections made automatically to incorrect or dated access control policies because nodes just enforce the policies signed by the owner (Egorov et al.). These problems that arise reflect the trade-offs that come from removing central authorities, as while the system gains control and resilience, it does so by placing the responsibility on policy management, key protection, and honest operations of the threshold of nodes.

Open Challenges and Future Research

There has been rapid progress in decentralized key management, despite which there are several open challenges that remain unresolved. The most pressing would be the lack of recovery methods that are truly user-friendly and do not introduce a central authority. As shown in NuCypher's KMS and the systems talked about earlier, the loss of a private key would mean loss

of access irreversibly as no node or server would be able to grant that again. There are recovery methods that grant partial solutions but they also require depending on having the correct configurations and trust in a distributed set of guardians which still opens risk to human errors. Another obstacle that comes up is issues with scaling the system, especially when considering the IoT examples in research by Dammak et al. and Ni et al. which highlight the immense communication and burden of tracking states it takes when having blockchain key updates. This becomes a problem when it comes to group membership that is largely or always changing. These systems would be operating securely, but the cost becomes prohibitive to apply this model in high volume environments like healthcare, energy grids and IoT networks that run in real time (Deepa et al.; Ni et al.; Zhang et al.). Lastly, there is also the challenge of governance and having regulatory alignment, which is difficult to establish without a central authority in charge. These concerns are from self-sovereign identities (SSI's) governance discussions which questions how DID methods are defined, how revocation rules are updated, and who takes responsibility for cryptographic rules breaking (Preukschat and Reed). This becomes a larger problem, especially in a setting with a higher emphasis on regulations like healthcare and finance.

Such challenges make way for essential research directions to build on decentralized key management systems. One direction would be developing hybrid trust models, which would combine the administration capabilities of PKI with the decentralization of DIDs, especially for healthcare sectors where identity must be distributed but also compliant with regulations (Deepa et al.). Another direction to explore would be to improve the people part of key management since both SSI literature and and systems like NuCypher showed how key loss and usability barriers remain as obstacles despite the system providing strong cryptographic measures (Egorov et al.; Preukschat and Reed). Lastly, there also needs to be ways to coordinate larger group

membership and state tracking without reintroducing the central authority and keeping the load on the system at a manageable level (Ni et al.). These research pathways show that in decentralized key management, cryptography is strong but the deployment in reality would require solving governance, usability, and scalability constraints.

Conclusion

Secure key management is the foundation of every security system because it governs how identities are established, how trust is enforced, and how access is controlled. Centralized PKI has upheld this process for decades, but its dependence on centralized authorities and hierarchical trust creates structural weaknesses that become increasingly difficult to manage in large or dynamic environments. Decentralized key management managed to address that limitation by replacing institutional trust with cryptography that spreads the responsibility across multiple nodes. DIDs, blockchain key protocols and threshold cryptography showed how the work performed by certificate authorities can be delivered through consensus and proof of key ownership. These systems reduce the risks of single points of failure and give users more control in an architecture that allows for dynamism in the setup.

However, the paper explored how decentralized KMS does not remove risk, it transforms it. There is a reliance on user-controlled keys which creates usability and recovery problems. Governance problems without a central authority come up, and the threshold systems are based on honest nodes being willing to participate to maintain availability. These gaps in the trust models are exposed when the system is put to the test in reality. As research expands on this topic, decentralized systems would continue to evolve, and their success would depend on providing cryptographic security but also making systems more usable for real world settings.

Works Cited

Dammak, M., et al. “Decentralized Lightweight Group Key Management for Dynamic Access Control in IoT Environments.” IEEE Transactions on Network and Service Management, vol. 17, no. 3, 2020, pp. 1742–1757.

Deepa, N., et al. “Decentralized Healthcare Management System Using Blockchain to Secure Sensitive Medical Data for Users.” Blockchain Security in Cloud Computing, edited by Lei Chen et al., Springer International Publishing, 2021, pp. 265–282.

Egorov, M., et al. NuCypher KMS: Decentralized Key Management System. arXiv:1707.06140, 2017, https://arxiv.org/pdf/1707.06140.

“NuCypher KMS: Decentralized Key Management System.” Medium, 2019, https://medium.com/nucypher/nucypher-kms-decentralized-key-management-system-7783cdaad39e.

Kim, Geun-Hyung. “A Comprehensive Survey of Cryptography Key Management in Decentralized Identity Ecosystem.” International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE), Springer Nature Singapore, 2025, pp. 128–140.

Ni, J., et al. “Distributed Group Key Management Based on Blockchain.” Electronics, vol. 13, no. 11, 2024, p. 2216, https://doi.org/10.3390/electronics13112216.

Preukschat, Alex, and Drummond Reed. Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials. Manning Publications, 2021.

“Decentralized Identifiers (DIDs) v1.0.” W3C Recommendation, 3 Mar. 2022, https://www.w3.org/TR/did-core/.

Yildiz, M., and Ş. Bahtiyar. “A Novel Key Management Framework for Secure and Scalable Decentralized Identity Systems.” 2024 17th International Conference on Security of Information and Networks (SIN), IEEE, 2024, pp. 1–8.

Zhang, H., J. Wang, and Y. Ding. “Blockchain-Based Decentralized and Secure Keyless Signature Scheme for Smart Grid.” Energy, vol. 180, 2019, pp. 955–967.