Achieving Proof of Knowledge through Holographic Witness Accumulation

Dylan Kawalec
Dr. Lin Wang
Dylan Rosario
Louis Lau
David Berger
Dr. Anish Mohammad
Dr. D.M. Lipscomb

July 2024

Abstract

This paper introduces the Enigma proof, a patent-pending proof system that employs a novel Holographic Morphism with Private Languages. The Enigma Cypher addresses the challenges of traditional cryptographic systems, which often rely on complex and computationally intensive operations. Unlike methods dependent on intricate Merkle trees and collision-resistant hash functions, our proposed Enigma Cypher focuses on human-compatible and interactive zero-knowledge proofs, enabling efficient "in-the-head" membership verifications on insecure channels or compromised devices. This innovative approach not only reduces cognitive load but also optimizes security in dynamic environments where privacy and security are paramount.

$$\mathcal{L}_S \cap \mathcal{L}_W = \emptyset \text{ and } \mathcal{L}_S \cup \mathcal{L}_W = \emptyset$$

The Enigma Cypher utilizes a unique set of enumerated alphabets with equal car-dinality, establishing a hidden morphism between two alphabet sets without union or intersection. This facilitates the creation of lower-dimensional, efficient, and se-cure witnesses. The holographic morphism maps higher-dimensional members of an alphabet into a newly formed lower-dimensional projective set, distributed in a way that supports dynamic, non-deterministic shuffling of set members. The higher-dimensional set contains a larger number of members, representing the individual components of the secret key, from which the witness is provided in any i^{th} round of a sigma protocol.

$$\text{Acc}_X = \{\Pi = (v_{i_1}, \ldots, v_{i_r}), \Theta = (v_{j_1}, \ldots, v_{j_r})\}$$

Furthermore, the Enigma Accumulator significantly enhances the scalability and security of cryptographic applications. It leverages Holographic Dimensional Non-Deterministic Language distributions over segmented alphabet projections to provide a more accessible and interactive proof system. This design makes the Enigma Cypher ideal for applications requiring robust security measures without the complexity and overhead associated with conventional methods, marking a substantial advancement in the field of cryptographic accumulators.

Introduction

A basic cryptographic accumulator Cypher facilitates optimal verification methods for set-membership relations. Given a set $\Xi = \{\xi_1, \ldots, \xi_n\}$, an accumulator Cypher simultaneously performs: (1) Accumulation, which produces a short representation of Ξ denoted as Acc_Ξ, and (2) Membership Witness Generation, which for every $\xi \in \Xi$, produces an accompanying short membership witness wit_ξ. Later, by exhibiting the valid tuple (ξ, wit_ξ, Acc_Ξ), a prover can convince any third party that ξ is indeed a member of the set represented by Acc_Ξ. This paper replaces Merkle trees with the Enigma Holographic Morphism with Private Languages to optimize witness generation and verification.

Our Contributions

This paper proposes an accumulator Cypher based on the Enigma Holographic Mor-phism with Private Languages. The Enigma Cypher introduces a method to generate efficient witnesses by using a set of enumerated alphabets with equal cardinality. There exists a hidden morphism between the two sets of alphabets, ensuring they have no union or intersection. This method leverages a projective set that contains within each of its members a fully dynamic, non-deterministic, randomly shuffled set. Once these members have been distributed to an arbitrary projective set, this set becomes a lower-dimensional representation of another alphabet, which itself has been segmented and its members distributed into the smaller projective set.

A straightforward application of accumulators is implementing membership testing systems such as authenticated dictionaries. These systems involve three parties: a trusted source, an untrusted directory, and a user. The source defines a finite set $\Xi \subseteq \mathcal{M}$ of elements. A short representation Acc_Ξ of Ξ is published, and users can obtain it in an authenticated manner. The directory maintains the sets $\{\mathrm{wit}_\xi \mid \xi \in \Xi\}$ and $\{\mathrm{wit}_\xi \mid \xi \in \mathcal{M} \backslash \Xi\}$. The user performs membership queries on the set Ξ by ask-ing, "Is element ξ in set Ξ?" To experience faster response times and avoid network latency, users query the directory instead of contacting the source directly. The di-rectory provides a yes/ no answer along with a witness wit_ξ, which yields proof of the answer. The user then verifies the proof based on ξ, wit_ξ, and Acc_Ξ.

Another typical membership testing system is a plain authentication system where the parties involved are users and a resource-carrying system holding all user credentials. The set Ξ defines the collection of all user identities. The resource system stores Acc_Ξ. Witnesses wit_ξ are distributed to each user $\xi \in \Xi$. Later, to access the resource system, a user can prove membership in Ξ by revealing their identity ξ and the witness wit_ξ. In both applications, the witnesses are continuously moved around, whereas the accumulation data Acc_Ξ remains static.

Traditional number-theoretic accumulators have proven to be better choices for these systems compared to simple Merkle tree-based accumulators, which make these systems communication-heavy due to increased witness sizes. However, an immediate question arises: can we have an accumulator Cypher that retains the advantages of simpler constructions while optimizing performance?

To address this, we propose an accumulator Cypher using the patented Enigma Holographic Morphism with Private Languages. Our Cypher introduces a novel method

to efficiently generate witnesses by using a set of enumerated alphabets with equal cardinality. There exists a hidden morphism between the two sets of alphabets, ensuring they have no union or intersection. This approach leverages a projective set that contains within each of its members a fully dynamic, non-deterministic, randomly shuffled set. Once these members have been distributed to an arbitrary projective set, this set becomes a lower-dimensional representation of another alphabet, which itself has been segmented and its members distributed into the smaller projective set.

1. **New Accumulator Cypher**: Introduces the Patented Enigma Holographic Mor-phism with private Languages for witness generation.
2. **Efficient Witness Generation**: Uses a set of enumerated alphabets with equal cardinality to optimize witness generation.
3. **Hidden Morphism**: Establishes a hidden morphism between two sets of alphabets with no union or intersection.
4. **Projective Set Utilization**: Leverages a projective set that contains within each member a dynamic, non-deterministic shuffled set.
5. **Lower-Dimensional Representation**: Distributes the members of a segmented alphabet into a smaller projective set, representing a lower-dimensional alphabet.

Our approach ensures that the optimization of the witness generation process is practical and beneficial for systems with memory-constrained devices, such as smart cards. This represents a significant advancement over traditional Merkle tree-based accumulators, offering a more efficient and scalable solution for modern cryptographic applications.

Preliminaries : This section covers essential concepts such as collision-resistant hash families and introduces the Enigma Holographic Morphism, which replaces tradi-tional Merkle trees. Definitions and properties of projective sets, subset covering, and the subset difference method are detailed, providing the foundation for the proposed accumulator Cypher.

Accumulator : This section details the proposed universal accumulator Cypher using the Enigma Holographic Morphism. The Cypher involves setting up the mor-phism, accumulating values, generating witnesses, and verifying membership/ non-membership.

Setup: In the setup phase, we define the projective set Π to model the domain $\mathcal{M}$. A synonym whole relation to the emergent set is established via a hidden morphism. Auxiliary information includes the morphism and a hash function $\text{aux}_M = \{\text{Morph}, H\}$.

Accumulate: The accumulate phase creates the short representation Acc_Ξ of the set Ξ, partitioning it into subsets that form the basis of the projective set:

$$\text{Acc}_\Xi = \{\Pi = (\nu_{i_1}, \ldots, \nu_{i_r}), \Theta = (\nu_{j_1}, \ldots, \nu_{j_r})\}$$

Witness Generation (WitGen)

For membership witness generation, a lower-dimensional reference is created using the Enigma Holographic Morphism. This reference is expanded to identify a subset of the projective alphabet sets:

$$\text{wit}_{\xi} = \mathcal{H}(\nu_{i_k}) \quad \text{for} \quad \xi \in \Xi$$

Non-membership witnesses are generated similarly, using lower-dimensional references expanded to verify non-inclusion in the set Ξ:

$$\text{wit}_{\xi'} = \mathcal{H}(\nu_{j_k}) \quad \text{for} \quad \xi' \in \mathcal{M} \setminus \Xi$$

Verification

The verification process utilizes the projective morphism to verify the exact path from the lower-dimensional witness to the accumulated representation Acc_{Ξ}. This ensures the validity of the witness in confirming membership or non-membership:

$$\text{Verify}(\text{resp}(\xi, \text{wit}_{\xi}, \chi), \text{Acc}_{\Xi}) = \begin{cases} \text{True} & \text{if valid membership} \\ \text{False} & \text{otherwise} \end{cases}$$

Security

The security of the proposed Cypher is analyzed, proving it to be secure under the assumption of the Enigma Holographic Morphism. The proof shows that an adversary cannot produce both valid membership and non-membership witnesses for the same element:

$$\mathbb{P}\left[\text{Adversary finds valid } (\xi^*, \text{wit}_{\xi^*}, \text{Acc}_{\Xi^*})\right] \leq \text{negl}(\lambda)$$

Efficiency

This section discusses the efficiency of the proposed Cypher, particularly the performance of witness generation and verification. It highlights the benefits of the Enigma Holographic Morphism in creating efficient cryptographic schemes suitable for memory-constrained environments.

Summary

The paper concludes by highlighting the achievement of a Enigma Holographic Morphism-based universal accumulator Cypher. The proposed Cypher represents a significant advancement in cryptographic accumulators, offering a more efficient and scalable solution for modern applications. Future work will explore additional trade-offs and optimizations in this area.

Enigma Accumulator with Lower Dimensional Witnesses

The Enigma Accumulator Cypher employs a novel approach to cryptographic accu-mulators, leveraging the Enigma Holographic Morphism with private Languages. This Cypher uses a set of enumerated alphabets with equal cardinality to create witnesses, ensuring efficient and secure membership proofs.

Witnesses in the Enigma Cypher are lower-dimensional references that are expanded after determining the holographic morphism. This morphism identifies a subset of the projective alphabet sets containing character members of another alphabet, whose members are distributed equally across the new lower-dimensional projective set. The higher-dimensional set contains a larger number of members, representing the

individual members of the secret key, from which the witness is provided in any i^{th} round of a sigma protocol.

Key Definitions

The Enigma Accumulator Cypher defines the projective set Π as a set containing elements arranged in a lower-dimensional space, which expands to identify specific subsets through holographic morphisms. The holographic morphism $\mathcal{H}$ is a hidden morphism linking two sets of alphabets Σ and Γ with no union or intersection. This morphism is used to expand lower-dimensional witnesses. The lower-dimensional representation process maps higher-dimensional members of an alphabet into lower-dimensional projective sets.

Witnesses in the Enigma Cypher are lower-dimensional references that are expanded after determining the holographic morphism. This morphism identifies a subset of the projective alphabet sets containing character members of another alphabet, whose members are distributed equally across the new lower-dimensional projective set. The higher-dimensional set contains a larger number of members, representing the individual members of the secret key, from which the witness is provided in any i^{th} round of a sigma protocol, given by

$$\text{wit}_\chi = \text{HoloMorphism}(\nu_{i_k}) \quad \text{for} \quad \chi \in \Xi$$

The projective set Π contains elements arranged in a lower-dimensional space, which expands to identify specific subsets through holographic morphisms. The holographic morphism $\mathcal{H}$ is a hidden morphism linking two sets of alphabets with no union or intersection, used to expand lower-dimensional witnesses. The lower-dimensional representation maps higher-dimensional members of an alphabet into lower-dimensional projective sets, denoted by

$$\Pi = \{\nu_{i_1}, \nu_{i_2}, \ldots, \nu_{i_r}\}$$

The Setup

In the setup phase, the projective set Π is defined to model the domain M. A synonym whole relation to the emergent set is established via a hidden morphism. During the accumulation phase, the short representation Acc_Ξ of the set X is created, partitioning it into subsets that form the basis of the projective set, represented as

$$\text{Acc}_\Xi = \{\Pi = (\nu_{i_1}, \nu_{i_2}, \ldots, \nu_{i_r}), \Omega = (\nu_{j_1}, \nu_{j_2}, \ldots, \nu_{j_r})\}$$

For membership witness generation, a lower-dimensional reference is created using the Enigma Holographic Morphism. This reference is expanded to identify a subset of the projective alphabet sets. Non-membership witnesses are generated similarly, using lower-dimensional references expanded to verify non-inclusion in the set X, expressed as

$$\text{wit}_\chi = \text{HoloMorphism}(\nu_{i_k}) \quad \text{for} \quad \chi \in \Xi$$

.

The verification process utilizes the projective morphism to verify the exact path from the lower-dimensional witness to the accumulated representation Acc_Ξ. This ensures the validity of the witness in confirming membership or non-membership, which is represented as

$$\text{Verify}(\text{resp}(\chi, \text{wit}_\chi, \chi), \text{Acc}_\Xi) = \begin{cases} \text{True} & \text{if valid membership} \\ \text{False} & \text{otherwise} \end{cases}$$

The primary equations and formulations in the Enigma Accumulator Cypher include the accumulator representation, membership witness generation, non-membership witness generation, and the sigma protocol for witness verification. The accumu-lator representation involves partitioning the set into projective subsets, formulated as

$$\text{Acc}_\Xi = \{\Pi = (\nu_{i_1}, \nu_{i_2}, \ldots, \nu_{i_r}), \Omega = (\nu_{j_1}, \nu_{j_2}, \ldots, \nu_{j_r})\}$$

For generating membership witnesses, the Enigma Holographic Morphism expands lower-dimensional references, expressed as $\text{wit}_\chi = \text{HoloMorphism}(\nu_{i_k}) \quad \text{for} \quad \chi \in \Xi$. Non-membership witnesses are generated using a similar approach, ensuring the references verify non-inclusion, given by

$$(]\text{wit}_{\chi'} = \text{HoloMorphism}(\nu_{j_k}) \quad \text{for} \quad \chi' \in \mathcal{M} \setminus \Xi$$

In the sigma protocol, the witness verification involves a challenge-response mechanism. The challenge χ is a random value, and the response is the expanded holographic morphism. The challenge is $\chi \in \{0,1\}^n$ and the response is $\text{resp}(\chi, \text{wit}_\chi, \chi) = \text{Expand}(\text{HoloMorphism}(\nu_{i_k}), \chi)$. The verification is given by

$$\text{Verify}(\text{resp}(\chi, \text{wit}_\chi, \chi), \text{Acc}_\Xi) = \begin{cases} \text{True} & \text{if valid membership} \\ \text{False} & \text{otherwise} \end{cases}$$

Consider a set $X = \{\chi_1, X_2, X_3, X_4\}$. The Enigma Accumulator would represent this set in a projective form, using lower-dimensional references expanded through the holographic morphism. The initial setup is $\text{Acc}_X = \{\Pi = (v_1, v_2, v_3, v_4), \Omega = (v_5, v_6, v_7, v_8)\}$. For membership witness generation for χ_1, we have $\text{wit}_{X_1} = \text{HoloMorphism}(v_1)$. The verification is conducted as $\text{Verify}(\text{resp}(X_1, \text{wit}_{X_1}, \chi), \text{Acc}_X)$

Setup and Accumulation

In the setup phase, the projective set Π is defined t o m odel t he d omain $\mathcal{M}$. A synonym whole relation to the emergent set is established via a hidden morphism. The accumulate phase creates the short representation Acc_Ξ of the set Ξ, partitioning it into subsets that form the basis of the projective set.

$$\text{Acc}_\Xi = \{\Pi = (\nu_{i_1}, \ldots, \nu_{i_r}), \Theta = (\nu_{j_1}, \ldots, \nu_{j_r})\}$$

Witness Generation

Membership witnesses are generated by creating a lower-dimensional reference using the Enigma Holographic Morphism $\mathcal{H}$. This reference is expanded to identify a subset of the projective alphabet sets. Non-membership witnesses are similar but verify non-inclusion in the set Ξ.

$$\text{wit}_\xi = \mathcal{H}(\nu_{i_k}) \quad \text{for} \quad \xi \in \Xi$$

$$\text{wit}_{\xi'} = \mathcal{H}(\nu_{j_k}) \quad \text{for} \quad \xi' \in \mathcal{M} \setminus \Xi$$

Verification

The verification process utilizes the projective morphism to verify the exact path from the lower-dimensional witness to the accumulated representation Acc_Ξ. This ensures the validity of the witness in confirming membership or non-membership.

$$\text{Verify}(\text{resp}(\xi, \text{wit}_\xi, \chi), \text{Acc}_\Xi) = \begin{cases} \text{True} & \text{if valid membership} \\ \text{False} & \text{otherwise} \end{cases}$$

Equations and Formulations

The primary equations and formulations in the Enigma Accumulator Cypher are:

1. **Accumulator Representation**:

$$\text{Acc}_\Xi = \{\Pi = (\nu_{i_1}, \ldots, \nu_{i_r}), \Theta = (\nu_{j_1}, \ldots, \nu_{j_r})\}$$

2. **Membership Witness Generation**:

$$\text{wit}_\xi = \mathcal{H}(\nu_{i_k}) \quad \text{for} \quad \xi \in \Xi$$

3. **Non-Membership Witness Generation**:

$$\text{wit}_{\xi'} = \mathcal{H}(\nu_{j_k}) \quad \text{for} \quad \xi' \in \mathcal{M} \setminus \Xi$$

4. **Sigma Protocol for Witness Verification**:

   - **Challenge**: $\chi \in \{0,1\}^n$
   - **Response**:
   $$\text{resp}(\xi, \text{wit}_\xi, \chi) = \text{Expand}(\mathcal{H}(\nu_{i_k}), \chi)$$
   - **Verification**:
   $$\text{Verify}(\text{resp}(\xi, \text{wit}_\xi, \chi), \text{Acc}_\Xi) = \begin{cases} \text{True} & \text{if valid membership} \\ \text{False} & \text{otherwise} \end{cases}$$

Example: Enigma Holographic Morphism

Consider a set $\Xi = \{\xi_1, \xi_2, \xi_3, \xi_4\}$. The Enigma Accumulator would represent this set in a projective form, using lower-dimensional references expanded through the holographic morphism:

- **Initial Setup**:

$$\text{Acc}_\Xi = \{\Pi = (\nu_1, \nu_2, \nu_3, \nu_4), \Theta = (\nu_5, \nu_6, \nu_7, \nu_8)\}$$

- **Membership Witness for ξ_1**:

$$\text{wit}_{\xi_1} = \mathcal{H}(\nu_1)$$

- **Verification**:
$$\text{Verify}(\text{resp}(\xi_1, \text{wit}_{\xi_1}, \chi), \text{Acc}_\Xi)$$

A Universal Accumulator Scheme via Enigma Cypher

We now present our scheme. The parameters of our scheme involve a security parameter $\lambda \in \mathbb{N}$, a message set $M = \{x_0, \ldots, x_{N-1}\}$ (we assume $N = 2^d$ for some $d \in \mathbb{N}$), and a family $\mathcal{H}_\lambda = \{H_k : \{0,\ 1\}^* \to \{0,\ 1\}^\lambda\}$ of λ-secure collision-resistant hash functions.

The scheme is described as follows:

SetUp (1^λ): Given λ and $M = \{x_0, \ldots, x_{N-1}\}$ as inputs, it proceeds as follows: Sample a λ-secure collision-resistant hash function $H = H_k : \{0,1\}^* \to \{0,1\}^\lambda \in \mathcal{H}_\lambda$. Define the projective set Π modeling the domain M. A synonym whole relation to the emergent set is established via a hidden morphism. The auxiliary information aux_M will be the input to both Accumulate and WitGen algorithms. **Accumulate** $(X \subseteq M, \text{aux}_M)$: For an arbitrary set $X \subseteq M$, its accumulation value Acc_X is generated as follows: Let $X = \{x_{j_0}, \ldots, x_{j_{t-1}}\}$ for some $j_0, \ldots, j_{t-1} \in \{0, \ldots, N-1\}$. Using aux_M, find i_k in $2^d \le i_k \le 2^{d+1} - 1 (k = 0, \ldots, t-1)$ such that $v_{i_k} = H(x_{j_k}) \in \Pi,\ k = 0, \ldots, t-1$. Set $X' = \{v_{i_0}, \ldots, v_{i_{t-1}}\}$ and $R' = \Pi \backslash X'$. The accumulation of X is set to:

$$\text{Acc}_X = \{\Pi = (v_{i_1}, \ldots, v_{i_r}), \Theta = (v_{j_1}, \ldots, v_{j_r})\}$$

WitGen $(x \in M, X \subseteq M, \text{Acc}_X, \text{aux}_M)$: On input $x \in M$, it computes a witness as follows:

(Case 1) $x \in X$: In this case, a membership witness is issued. There exists a unique $(i,j) \in \Pi$ for which $S_{i,j}$ has a leaf node $v_\nu = H(x)$, where $2^d \le \nu < 2^{d+1}$. The membership witness is set to the exact path from v_ν to v_i.

$$\text{wit}_x = \mathcal{H}(\nu_{i_k}) \quad \text{for} \quad x \in X$$

(Case 2) $x \in M \backslash X$: In this case, a non-membership witness is issued. There exists a unique $S_{i,j} \in \Pi$ such that T_j has the leaf node $v_\nu = H(x)$, where $2^d \le \nu < 2^{d+1}$ and $\Pi \cap X' = \emptyset$. The non-membership witness is set to the exact path from v_ν to v_j.

$$\text{wit}_{x'} = \mathcal{H}(\nu_{j_k}) \quad \text{for} \quad x' \in M \setminus X$$

Witness Generation of Non-Deterministic Alphabets

The witness generation process reflects the use of a non-deterministic random stochastic distribution of the alphabet, from which the secret key's members are derived. This distribution ensures that the members of the secret alphabet are evenly distributed within a segmented subset of the projective witness alphabet, achieving both confusion and diffusion as described by Claude Shannon. The Enigma Cypher utilizes a zero-knowledge proof mechanism, ensuring that the private morphism remains hidden while providing verifiable proof of membership or non-membership.

Verify $(x, \text{wit}_x, \text{Acc}_X)$: Suppose, $\text{wit}_x = ((v_{\theta_\ell}, \tau_\ell), (v_{\theta_{\ell-1}}, \tau_{\ell-1}), \ldots, (v_{\theta_1}, \tau_1))$, $\ell \le d$. The verification algorithm proceeds as follows: Let $V_\ell = H(x)$. It computes the exact path from V_ℓ to a node in Π as follows. Recursively compute V_i's, $i = \ell - 1, \ldots, 0$, the internal nodes on the exact path from V_ℓ to this node as follows:

$$V_i = \begin{cases} H(V_{i+1}, v_{\theta_{i+1}}) & \tau_{i+1} = -1 \\ H(v_{\theta_{i+1}}, V_{i+1}) & \tau_{i+1} = 1 \end{cases}$$

Thus, $\mathrm{EP}_{V_\ell \to V_0} = (V_\ell, V_{\ell-1}, \ldots, V_1, V_0)$. The algorithm finally outputs "mem" /"non-mem" as follows:

- Case 1: $V_0 = v_{i_k} \in$ mem for some k in $1 \leq k \leq r$.

$$\text{Output} = \begin{cases} \perp, \text{if } \exists \text{ an } \eta \text{ in } 1 \leq \eta \leq \ell - 1 \text{ with } V_\eta \in \text{non-mem} \\ \text{mem, otherwise} \end{cases}$$

- Case 2: $V_0 = v_{j_k} \in$ non-mem for some k in $1 \leq k \leq r$.

$$\text{Output} = \begin{cases} \perp, \text{if } \exists \text{ an } \eta \text{ in } 1 \leq \eta \text{ with } V_\eta \in \text{mem} \\ \text{non-mem, otherwise} \end{cases}$$

- Output $\perp$, otherwise.

Efficiency

The primary motivation behind this work is to find a way that takes us beyond the logarithmic size bottleneck for witnesses, which is typical for Merkle tree-based accumulator schemes. In the Enigma Cypher, the witnesses are evenly distributed within a segmented subset of the projective witness alphabet. This ensures that the witness generation achieves both confusion and diffusion, enhancing security while maintaining efficiency. Unlike existing schemes, the Enigma Cypher maintains the same level of security with more efficient witness generation and verification processes. The use of non-deterministic random stochastic distribution ensures that the members of the secret alphabet are evenly distributed, leading to robust and secure accumulator schemes suitable for modern cryptographic applications.

Proofs

Theorem 1: Collision-Resistance of Hash Functions

For all domain sets M, all $\lambda \in \mathbb{N}$, and for all polynomial time (in λ) adversaries $\mathcal{A}$, there exists a negligible function $\mathrm{negl}(\lambda)$ such that:

$$\mathbb{P}[\mathcal{A}(1^\lambda) \to (x_1, x_2) \text{ such that } H(x_1) = H(x_2)] \leq \mathrm{negl}(\lambda)$$

where the probability $\mathbb{P}[\cdot]$ is computed over randomness in the Setup algorithm.

Theorem 2: Correctness of Membership and Non-Membership Witnesses

For any $x \in M, X \subseteq M$, an adversary cannot simultaneously produce both membership and non-membership witnesses. The proof below considers the following two cases:

Case 1: Assume $x \in X$. Thus $H(x) \in X'$. As $CV_{R'}$ partitions X', there exists a unique $S_{i,j} \in CV_{R'}$ such that $H(x) \in S_{i,j} = \mathrm{LN}(T_i \backslash T_j)$. Therefore, a valid membership witness for x exists and it is equal to $\mathrm{PP}_{H(x) \to v_i}$, where v_i is the root node for subtree T_i. We now show that a non-membership witness for x cannot be issued simultaneously. Assuming otherwise, let wit'_x be such that (Verify(x, wit_x', Acc_X) = non-mem). This implies $\mathrm{wit}'_x = \mathrm{PP}_{H(x) \to v_k}$, where $v_k \in$ Non-Mem and $\mathrm{EP}_{H(x) \to v_k}$ doesn't have any internal node belonging to Mem. Clearly, $k \triangleright i$. Therefore, $k \leq i$.

But this implies that the exact path $\mathrm{EP}_{H(x)\to v_k}$, computed using $\mathrm{PP}_{H(x)\to v_k}$, must contain v_i as an internal node. This is true as $H(x)$ is a leaf node of T_i. This is a contradiction as $v_i \in$ Mem and therefore $\mathrm{Verify}(x, \mathrm{wit}'_x, \mathrm{Acc}_X)$ will output $\perp$.

Case 2: Assume $x \in M \backslash X$. There exists a unique $S_{i,j} \in CV_{R'}$ such that $H(x)$ is the leaf node of T_j and $\mathrm{LN}(T_j) \cap X' = \emptyset$. Therefore, a valid non-membership witness for x exists and it is equal to $\mathrm{PP}_{H(x)\to v_j}$, where v_j is the root node for subtree T_j. We now show that a membership witness for x cannot be issued simultaneously. Assuming otherwise, let wit_x be a membership witness for x. This implies $\mathrm{wit}_x = \mathrm{PP}_{H(x)\to v_k}$, where $v_k \in$ Mem and the exact path $\mathrm{EP}_{H(x)\to v_k}$ doesn't have any internal node belonging to Non-Mem. Clearly, $k = i$ will not work. For any other choice, $k \geq j$ as $\mathrm{LN}(T_j) \cap X' = \emptyset$. Therefore, $k < j$. But this implies that the exact path $\mathrm{EP}_{H(x)\to v_k}$, computed using $\mathrm{PP}_{H(x)\to v_k}$, must contain v_j as an internal node. This is true as $H(x)$ is a leaf node of T_j. This is a contradiction as $v_j \in$ Non-Mem and therefore $\mathrm{Verify}(x, \mathrm{wit}_x, \mathrm{Acc}_X)$ will output $\perp$.

Theorem 3: Zero-Knowledge Property of Enigma **Cypher**

The Enigma Cypher ensures zero-knowledge proof of membership or non-membership without revealing the actual value or the private morphism used.
Statement: The zero-knowledge property is achieved by the holographic morphism $\mathcal{H}$, which transforms the elements into a different domain while preserving the ability to verify membership or non-membership. The key aspects are:

1. **Holographic Transformation**: The transformation $\mathcal{H}$ maps elements to a projective set Π such that the actual value remains hidden.
2. **Non-Deterministic Distribution**: The distribution of the alphabet ensures that the witness does not reveal any information about the original set X.

During the verification process, only the transformed values are used, and the original values and morphism remain concealed. This ensures that the verifier learns nothing beyond the validity of the witness, satisfying the zero-knowledge requirement:

$$\text{Verifier learns only if } x \in X \text{ or } x \notin X \text{ without additional information about } x$$

These proofs validate the robustness, correctness, and security of the Enigma Cypher scheme, making it a reliable method for secure accumulators and witness generation.

Theorem 4: Zero-Knowledge Property of Enigma **Cypher**

The Enigma Cypher ensures zero-knowledge proof of membership or non-membership without revealing the actual value or the private morphism used.
Statement: The zero-knowledge property is achieved by the holographic morphism $\mathcal{H}$, which transforms the elements into a different domain while preserving the ability to verify membership or non-membership. The key aspects are:

1. **Holographic Transformation**: The transformation $\mathcal{H}$ maps elements to a projective set Π such that the actual value remains hidden. Given an element $x \in X$, $\mathcal{H}$ maps it to $v_\nu \in \Pi$, maintaining the secrecy of x while allowing verification through the witness.

$$\mathcal{H}(x) = v_\nu \quad \text{for} \quad x \in X$$

2. **Non-Deterministic Distribution**: The distribution of the alphabet ensures that the members of the secret key's derived language are not members of the witness language, embodying Gödel's incompleteness theorem. This theorem states that there is no union or intersection of the bijective sets due to the hidden morphism between the sets. This characteristic is crucial in the Enigma Cypher, as it uses private statements of truth to ensure that proof of knowledge can still be achieved without leaking any information.

3. **Proof of Knowledge**: During the verification process, only the transformed values are used, and the original values and morphism remain concealed. This ensures that the verifier learns nothing beyond the validity of the witness, satisfying the zero-knowledge requirement.

Verifier learns only if $x \in X$ or $x \notin X$ without additional information about x

The complexity of attacking a projective secret key derived language whose members are not a member of the witness language is exponentially hard due to the hidden morphism $\mathcal{H}$. This hidden morphism ensures that the projective set Π and the secret key's derived language have no union or intersection, making it computationally infeasible for an adversary to derive the original elements from the transformed values. **Gödel's Incompleteness Theorem in Enigma Cypher**: The Enigma Cypher lever-ages Gödel's incompleteness theorem by ensuring that the secret key's derived lan-guage ($\mathcal{L}_S$) and the witness language ($\mathcal{L}_W$) have no union or intersection. This is achieved through the hidden morphism $\mathcal{H}$, which guarantees that:

$$\mathcal{L}_S \cap \mathcal{L}_W = \emptyset \text{ and } \mathcal{L}_S \cup \mathcal{L}_W = \emptyset$$

These bijective sets are mapped such that their elements cannot be simultaneously members of both languages. This property is essential for maintaining the zero-knowledge proof, as it ensures that the knowledge of membership or non-membership can be verified without revealing any additional information about the secret keys or the private morphism.

Proofs

1. **Holographic Transformation**:

   - The holographic morphism $\mathcal{H}$ maps elements x from the set X to a projective set Π, ensuring that the actual value x remains hidden while allowing verification through the witness v_ν.

$$\mathcal{H}(x) = v_\nu \quad \text{for} \quad x \in X$$

2. **Non-Deterministic Distribution**:

- The distribution of the alphabet ensures that the members of the secret key's derived language ($\mathcal{L}_S$) are not members of the witness language ($\mathcal{L}_W$), leveraging Gödel's incompleteness theorem. This means there is no union or intersection of these bijective sets due to the hidden morphism $\mathcal{H}$.

$$\mathcal{L}_S \cap \mathcal{L}_W = \emptyset \quad \text{and} \quad \mathcal{L}_S \cup \mathcal{L}_W = \emptyset$$

3. **Proof of Knowledge**:

   - During the verification process, only the transformed values are used, keeping the original values and morphism concealed. The verifier learns only whether $x \in X$ or $x \notin X$ without gaining additional information about x.

Verifier learns only if $x \in X$ or $x \notin X$ without additional information about x

4. **Security through Hidden Morphism**:

   - The complexity of attacking a projective secret key derived language ($\mathcal{L}_S$) whose members are not in the witness language ($\mathcal{L}_W$) is exponentially hard due to the hidden morphism $\mathcal{H}$. This ensures that the projective set Π and the secret key's derived language have no union or intersection, making it computationally infeasible for an adversary to derive the original elements from the transformed values.

$$\mathcal{L}_S \cap \mathcal{L}_W = \emptyset \quad \text{and} \quad \mathcal{L}_S \cup \mathcal{L}_W = \emptyset$$

4.2 Definitions o f Proof

To illustrate the robustness of our scheme, we have present a set of equations representing the principal schema and protocol for sigma challenge and response with witnesses in the form of holographic alphabet members. These members act as synonyms for a lower-dimensional collapsed secret key alphabet containing the members from which the secret key was derived.

4.3 Gödel's Incompleteness Cypher Proof

$$\mathcal{H}(x) = v_\nu \quad \text{for} \quad x \in X \quad \text{and} \quad \mathcal{L}_S \cap \mathcal{L}_W = \emptyset \quad \text{and} \quad \mathcal{L}_S \cup \mathcal{L}_W = \emptyset$$

$$\forall s \in \mathcal{L}_S, \forall w \in \mathcal{L}_W, s \neq w$$

Gödel's incompleteness theorem states that in any consistent formal system, there are statements that are true but cannot be proven within the system. The Enigma Cypher leverages this concept by ensuring that there is no union or intersection of the bijective sets due to the hidden morphism between the sets. This hidden morphism creates a scenario where the elements of the secret alphabet and the projective witness alphabet remain distinct and non-overlapping. Consequently, even if an adversary gains partial knowledge, they cannot derive the complete structure or key, aligning with the principles of Gödel's theorem.

Holographic Transformation Function

$$\mathcal{H}(x) = v_\nu \quad \text{for} \quad x \in X$$

- Let $\mathcal{H}$ denote a linear transformation matrix.
- Let x denote a vector from the vector space X.
- Let X denote the vector space of all possible input vectors.
- Let v_ν denote the resulting vector in the projective space Π after applying the linear transformation $\mathcal{H}$.

This equation states that when the linear transformation $\mathcal{H}$ is applied to a vector x from the vector space X, it maps to a vector v_ν in the projective space Π. This transformation matrix ensures that each input vector is uniquely mapped to a projective vector, maintaining the integrity and security of the cryptographic process.

Non-Intersecting and Non-Union Sets

$$\mathcal{L}_S \cap \mathcal{L}_W = \emptyset \quad \text{and} \quad \mathcal{L}_S \cup \mathcal{L}_W = \emptyset$$

- Let $\mathcal{L}_S$ denote the secret alphabet vector space.
- Let $\mathcal{L}_W$ denote the projective witness alphabet vector space.
- Let $\cap$ denote the intersection operator, representing the common vectors between two vector spaces.
- Let $\cup$ denote the union operator, representing all vectors from both vector spaces combined.
- Let $\emptyset$ denote the empty vector space, indicating no vectors.

This pair of equations states that the secret alphabet vector space $\mathcal{L}_S$ and the projective witness alphabet vector space $\mathcal{L}_W$ have no common vectors (intersection is empty) and that their union is also empty, meaning they are distinct and non-overlapping. This distinction is crucial for maintaining the security of the cypher, as it prevents any overlap or commonality between the vector spaces that could be exploited by an adversary.

Perfect Secrecy of the Enigma Cypher

The Enigma Cypher achieves perfect secrecy through its unique approach to crypto-graphic accumulators using the Enigma Holographic Morphism. This morphism en-sures that members of the secret alphabet are evenly distributed within a segmented subset of the projective witness alphabet, achieving both confusion and diffusion as described by Claude Shannon. This approach guarantees that proof of knowledge can be achieved without leaking any information, maintaining the privacy and security of the cryptographic protocol.

Distinct Elements in Sets

$$\forall s \in \mathcal{L}_S, \forall w \in \mathcal{L}_W, s \neq w$$

- Let $\forall$ denote the universal quantifier, meaning "for all."
- Let s denote a vector from the secret alphabet vector space $\mathcal{L}_S$.
- Let w denote a vector from the projective witness alphabet vector space $\mathcal{L}_W$.
- Let $\neq$ denote the inequality operator, meaning "not equal to."

This equation states that for all vectors s in the secret alphabet vector space $\mathcal{L}_S$ and w in the projective witness alphabet vector space $\mathcal{L}_W$, s is not equal to w. This reinforces the distinctness of the vector spaces, ensuring that each vector in the secret alphabet is unique and does not appear in the projective witness alphabet, thus enhancing the security of the cypher.

Shannon Confusion and Diffusion

$$\mathcal{D}(\mathcal{L}_S, \mathcal{L}_W) = \{\nu_i \mid \nu_i \in \mathcal{L}_W, \forall i \in [1, |\mathcal{L}_W|]\}$$

$$\mathcal{H}(\mathcal{L}_S) \neq \mathcal{L}_W \text{and} \, \forall x \in \mathcal{L}_S \, , \mathcal{H}(x) \, \text{spreads over} \, \mathcal{L}_W$$

The Enigma Cypher adheres to Claude Shannon's principles of confusion and diffu-sion, which are fundamental in cryptographic design.

- **Confusion**: The holographic morphism transforms elements from the secret alphabet into a different domain (projective witness alphabet), hiding the actual values and obfuscating the direct relationship between the input and output, thereby increasing the complexity for any adversary attempting to reverse-engineer the scheme.
- **Diffusion**: By distributing the members of the secret alphabet evenly across the projective witness alphabet, the Enigma Cypher ensures that the statistical structure of the secret key is thoroughly dispersed, making it much harder for an attacker to gather useful information from the ciphertext.

These principles are encapsulated in the following equations:

Distribution Function

$$\mathcal{D}(\mathcal{L}_S, \mathcal{L}_W) = \{\nu_i \mid \nu_i \in \mathcal{L}_W, \forall i \in [1, |\mathcal{L}_W|]\}$$

- Let $\mathcal{D}$ denote a distribution function.
- Let $\mathcal{L}_S$ denote the secret alphabet vector space.
- Let $\mathcal{L}_W$ denote the projective witness alphabet vector space.
- Let ν_i denote a vector of the projective witness alphabet vector space $\mathcal{L}_W$.
- Let $\forall i \in [1, |\mathcal{L}_W|]$ denote for all i in the range from 1 to the size of the vector space $\mathcal{L}_W$.

This equation describes how vectors from the secret alphabet $\mathcal{L}_S$ are distributed within the projective witness alphabet $\mathcal{L}_W$. The distribution function $\mathcal{D}$ ensures that each vector from the secret alphabet is mapped to a corresponding vector in the projective witness alphabet, achieving an even and secure distribution.

Transformation and Diffusion

$$\mathcal{H}(\mathcal{L}_S) \neq \mathcal{L}_W \quad \text{and} \quad \forall x \in \mathcal{L}_S, \mathcal{H}(x) \text{ spreads over } \mathcal{L}_W$$

- Let $\mathcal{H}$ denote a linear transformation matrix.
- Let $\mathcal{L}_S$ denote the secret alphabet vector space.
- Let $\mathcal{L}_W$ denote the projective witness alphabet vector space.
- Let x denote a vector from the secret alphabet vector space $\mathcal{L}_S$.

This pair of equations states that the linear transformation of the secret alphabet $\mathcal{L}_S$ results in vectors that are not in $\mathcal{L}_W$ and that each vector x in $\mathcal{L}_S$ is transformed and spread over the projective witness alphabet $\mathcal{L}_W$. This ensures that the transformation results in a thorough dispersion of vectors, achieving the diffusion necessary to obfuscate the statistical structure of the original data.

Proof of Knowledge

Verifier learns only if $x \in X$ or $x \notin X$ without additional information about x

Additionally, the computational infeasibility for adversaries to derive original elements is highlighted by:

$$\text{Attack Complexity: } \mathcal{O}(2^{|X|})$$

No-Information Leakage in Verification

Verifier learns only if $x \in X$ or $x \notin X$ without additional information about x

- Let x denote a vector being verified.
- Let X denote the vector space of all possible input vectors.

This statement means the verifier only learns whether a vector x is a member of the vector space X or not, without gaining any additional information about x. This process ensures that the verifier can confirm the presence or absence of a vector in the vector space without compromising the privacy or security of the vector being verified.

Attack Complexity

$$\text{Attack Complexity: } \mathcal{O}(2^{|X|})$$

- Let $\mathcal{O}$ denote Big-O notation, representing the upper bound of the algorithm's complexity.
- Let $|X|$ denote the dimension of the vector space X.

This equation describes the computational complexity of attacking the system, which is exponential in relation to the dimension of the vector space X. This high level of complexity indicates that an adversary would require a significant amount of computational resources to compromise the system, thereby ensuring the robustness and security of the cryptographic protocol.

Equation for Sigma Challenge and Response

Let $\mathcal{H}$ be the holographic morphism, ν_{ι_κ} represent members of the holographic alphabet, and Σ denote the secret key alphabet. The principal schema for the Enigma Cypher's sigma protocol can be expressed as follows:

1. **Holographic Transformation**:

$$\mathcal{H}(\chi) = \nu_\nu \quad \text{for} \quad \chi \in \Xi$$

2. **Witness Generation**:

$$\text{wit}_\chi = \mathcal{H}(\nu_{\iota_\kappa}) \quad \text{for} \quad \chi \in \Xi$$

$$\text{wit}_{\chi'} = \mathcal{H}(\nu_{j_\kappa}) \quad \text{for} \quad \chi' \in \mathcal{M} \setminus \Xi$$

3. **Sigma Protocol Challenge**:

$$\kappa \in \{0,1\}^\nu$$

4. **Sigma Protocol Response**:

$$\text{resp}(\chi, \text{wit}_\chi, \kappa) = \text{Expand}(\mathcal{H}(\nu_{\iota_\kappa}), \kappa)$$

5. **Verification**:

$$\nu_\iota = \begin{cases} H(\nu_{\iota+1}, \nu_{\theta_{\iota+1}}) & \tau_{\iota+1} = -1 \\ H(\nu_{\theta_{\iota+1}}, \nu_{\iota+1}) & \tau_{\iota+1} = 1 \end{cases}$$

$$\text{EP}_{\nu_\lambda \to \nu_0} = (\nu_\lambda, \nu_{\lambda-1}, \ldots, \nu_1, \nu_0)$$

6. **Final Verification Output**:

   - Case 1: $\nu_0 = \nu_{\iota_\kappa} \in$ mem for some κ in $1 \leq \kappa \leq \rho$.

$$\text{Output} = \begin{cases} \perp, \text{if } \exists \text{ an } \eta \text{ in } 1 \leq \eta \leq \lambda - 1 \text{ with } \nu_\eta \in \text{non-mem} \\ \text{mem, otherwise} \end{cases}$$

   - Case 2: $\nu_0 = \nu_{j_\kappa} \in$ non-mem for some κ in $1 \leq \kappa \leq \rho$.

$$\text{Output} = \begin{cases} \perp, \text{if } \exists \text{ an } \eta \text{ in } 1 \leq \eta \text{ with } \nu_\eta \in \text{mem} \\ \text{non-mem, otherwise} \end{cases}$$

   - Output $\perp$, otherwise.

By leveraging the holographic principle of information theory, we efficiently achieve zero knowledge proofs through cryptographic accumulator.

Accumulate

The accumulate phase creates the short representation Acc_Ξ of the set Ξ, partitioning it into subsets that form the basis of the projective set:

$$\text{Acc}_\Xi = \{\Pi = (\nu_{\iota_1}, \ldots, \nu_{\iota_\rho}), \Theta = (\nu_{j_1}, \ldots, \nu_{j_\rho})\}$$

Witness Generation (WitGen)

For membership witness generation, a lower-dimensional reference is created using the Enigma Holographic Morphism. This reference is expanded to identify a subset of the projective alphabet sets:

$$\text{wit}_{\xi} = \mathcal{H}(\nu_{\iota_{\kappa}}) \quad \text{for} \quad \xi \in \Xi$$

Non-membership witnesses are generated similarly, using lower-dimensional references expanded to verify non-inclusion in the set Ξ:

$$\text{wit}_{\xi'} = \mathcal{H}(\nu_{j_{\kappa}}) \quad \text{for} \quad \xi' \in \mathcal{M} \setminus \Xi$$

Verification

The verification process utilizes the projective morphism to verify the exact path from the lower-dimensional witness to the accumulated representation Acc_{Ξ}. This ensures the validity of the witness in confirming membership or non-membership:

$$\text{Verify}(\text{resp}(\xi, \text{wit}_{\xi}, \kappa), \text{Acc}_{\Xi}) = \begin{cases} \text{True} & \text{if valid membership} \\ \text{False} & \text{otherwise} \end{cases}$$

Non-Deterministic Random Stochastic Distribution

The Enigma Cypher employs a non-deterministic random stochastic distribution to distribute members of the secret alphabet evenly within a segmented subset of the projective witness alphabet. This even distribution is essential for maintaining the integrity and security of the cryptographic scheme, making it difficult for adversaries to infer patterns or predict the distribution of the elements.

Proof of Knowledge without Information Leakage

The Enigma Cypher achieves zero-knowledge proofs, ensuring that the verifier can be convinced of a statement's truth without gaining any additional information about the statement itself. This is accomplished through the following mechanisms:

- **Holographic Morphism**: Transforms elements to a projective set, concealing the actual values.
- **Non-Deterministic Distribution**: Ensures that the witness does not reveal any information about the original set.
- **Verification Process**: Uses transformed values while keeping the original values and morphism hidden.

These mechanisms ensure that the verifier learns only whether a given element is a member or non-member of a set, without gaining any additional information about the element itself, maintaining privacy and security in cryptographic protocols.

Conclusions

In this work, we have proposed the Enigma Cypher that utilizes a novel holographic morphism to map higher dimensional alphabets into a lower dimensional projective set Π, ensuring that the original values remain concealed while enabling the verifi-cation of membership or non-membership. The non-deterministic distribution of the alphabet ensures that the witness does not reveal any information about the original set. This approach, grounded in Claude Shannon's principles of confusion and diffu-sion, provides robust protection against cryptographic attacks.
In this novel and original work, we have proposed an approach to cryptographic accumulators leveraging the Enigma Holographic Morphism witness and dynamic manifold projections with private languages dimensional collapse. Our scheme ensures effi-cient and secure membership proofs without revealing the actual values or private morphisms used. By utilizing non-deterministic random stochastic distribution, our scheme ensures that members of the secret alphabet are evenly distributed within a segmented subset of the projective witness alphabet, achieving both confusion and diffusion as described by Claude Shannon.
The security of the Enigma Cypher is further bolstered by Gödel's incompleteness the-orem, ensuring that there is no union or intersection between the bijective sets due to the hidden morphism between them. This principle underpins the private statements of truth within the Enigma Cypher, enabling proof of knowledge.

This method embodies Gödel's incompleteness theorem, stating that there is no union or intersection of the bijective sets due to the hidden morphism between the sets. Thus, proof of knowledge can be achieved without leaking any information.

References

[**1**] Jae Hyun Ahn, Dan Boneh, Jan Camenisch, Susan Hohenberger, Abhi Shelat, and Brent Waters. Computing on authenticated data. In TCC, volume 7194 of Lecture Notes in Computer Science, pages 1-20. Springer, 2012.

[**2**] Tolga Acar and Lan Nguyen. Revocation for delegatable anonymous credentials. In PKC, volume 6571 of Lecture Notes in Computer Science, pages 423-440. Springer, 2011.

[**3**] Dan Boneh and Henry Corrigan-Gibbs. Bivariate polynomials modulo composites and their applications. In Palash Sarkar and Tetsu Iwata, editors, ASIACRYPT, volume 8873 of Lecture Notes in Computer Science, pages 42-62. Springer, 2014.

[**4**] Josh Cohen Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital signatures (extended abstract). In EUROCRYPT, volume 765 of Lecture Notes in Computer Science, pages 274-285. Springer, 1993.

[**5**] Ahto Buldas, Peeter Laud, and Helger Lipmaa. Accountable certificate management using undeniable attestations. In ACM CCS, pages 9-17, 2000.

[**6**] Ahto Buldas, Peeter Laud, and Helger Lipmaa. Eliminating counterevidence with applications to accountable certificate management. Journal of Computer Security, 10(3): 273-296, 2002.

[7] Niko Bari and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In EUROCRYPT, volume 1233 of Lecture Notes in Computer Science, pages 480-494. Springer, 1997.

[8] Dario Catalano and Dario Fiore. Vector commitments and their applications. In PKC, volume 7778 of Lecture Notes in Computer Science, pages 55-72. Springer, 2013.

[9] Philippe Camacho, Alejandro Hevia, Marcos Kiwi, and Roberto Opazo. Strong accumulators from collision-resistant hashing. International Journal of Information Security, 11(5):349-363, 2012.

[10] Sébastien Canard and Amandine Jambert. On extended sanitizable signature schemes. In CT-RSA, volume 5985 of Lecture Notes in Computer Science, pages 179-194. Springer, 2010.

[11] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In PKC, volume 5443 of Lecture Notes in Computer Science, pages 481-500. Springer, 2009.

[12] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In CRYPTO, volume 2442 of Lecture Notes in Computer Science, pages 61-76. Springer, 2002.

[13] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In EUROCRYPT, volume 3027 of Lecture Notes in Computer Science, pages 609-626. Springer, 2004.

[14] Ivan Damgård and Nikos Triandopoulos. Supporting non-membership proofs with bilinear-map accumulators. IACR Cryptology ePrint Archive, 2008:538, 2008.

[15] Esha Ghosh, Olga Ohrimenko, Dimitrios Papadopoulos, Roberto Tamassia, and Nikos Triandopoulos. Zero-knowledge accumulators and set algebra. In ASIACRYPT, volume 10032 of LNCS, pages 67-100, 2016.

[16] Michael T. Goodrich, Roberto Tamassia, and Jasminka Hasic. An efficient dynamic and distributed cryptographic accumulator. In ISC, volume 2433 of LNCS, pages 372-388. Springer, 2002.

[17] Mahabir Prasad Jhanwar and Reihaneh Safavi-Naini. Compact accumulator using lattices. In SPACE, volume 9354 of LNCS, pages 347-358. Springer, 2015.

[18] Helger Lipmaa. Secure accumulators from euclidean rings without trusted setup. In ACNS, volume 7341 of LNCS, pages 224-240. Springer, 2012.

[19] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In EUROCRYPT, volume 9666 of LNCS, pages 1-31. Springer, 2016.

[20] Jiangtao Li, Ninghui Li, and Rui Xue. Universal accumulators with efficient nonmembership proofs. In ACNS, Lecture Notes in Computer Science.

[**21**] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In IEEE Symposium on Security and Privacy. IEEE Computer Society.

[**22**] Lan Nguyen. Accumulators from bilinear pairings and applications. In CT-RSA, volume 3376 of Lecture Notes in Computer Science, pages 275-292. Springer, 2005.

[**23**] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, CRYPTO, volume 2139 of Lecture Notes in Computer Science, pages 41-62. Springer, 2001.

[**24**] Kaisa Nyberg. Fast accumulated hashing. In FSE, volume 1039 of Lecture Notes in Computer Science, pages 83-87. Springer, 1996.

[**25**] Henrich Christopher Pöhls and Kai Samelin. On updatable redactable signatures. In ACNS, volume 8479 of Lecture Notes in Computer Science, pages 457-475. Springer, 2014.

[**26**] Gene Tsudik and Shouhuai Xu. Accumulating composites and improved group signing. In ASIACRYPT, volume 2894 of LNCS, pages 269-286. Springer, 2003.

[**27**] Peishun Wang, Huaxiong Wang, and Josef Pieprzyk. Improvement of a dynamic accumulator at ICICS 07 and its application in multi-user keyword-based retrieval on encrypted data. In APSCC, pages 1381-1386. IEEE Computer Society, 2008.