Hypersphere Secure Sketch Revisited: Probabilistic Linear Regression Attack on IronMask in Multiple Usage

Pengxu Zhu
Shanghai Jiao Tong University
zhupengxu@sjtu.edu.com

Lei Wang*
Shanghai Jiao Tong University
wanglei_hb@sjtu.edu.cn

Abstract

Protection of biometric templates is a critical and urgent area of focus. **IronMask** demonstrates superior recognition performance while protecting facial templates against existing known attacks. In high-level, IronMask can be conceptualized as a fuzzy commitment scheme building on the hypersphere directly. We devise an attack on IronMask targeting on the security notion of renewability. Our attack, termed as **Probabilistic Linear Regression Attack**, utilizes the linearity of underlying used error correcting code. This attack is the first algorithm to successfully recover the original template when getting multiple protected templates in acceptable time and requirement of storage. We implement experiments on **IronMask** applied to protect **ArcFace** that well verify the validity of our attacks. Furthermore, we carry out experiments in noisy environments and confirm that our attacks are still applicable. Finally, we discuss two strategies to mitigate this type of attacks.

Keywords

biometric template protection, secure sketch, fuzzy commitment, security analysis, face recognition

1 Introduction

Biometric-based authentication has been under intensive and continuous investigation for decades. Recent works use deep neural networks to extract discriminative features from users' biometrics and achieve significant advances, such as facial images. **ArcFace** [12], which is one of the state-of-the-art face recognition system, projects the face images to templates on a hypersphere and utilizes angular distance to distinguish identities.

However, the exposure of facial templates has the potential to cause severe threats to both user privacy and the entire authentication system. Numerous attacks have demonstrated the risks of template leakage with some capable of reconstructing original biometrics from corresponding leaked templates, including faces [33, 37, 38], fingerprints [8], iris [15] and finger veins [25]. These vulnerabilities underscore the critical need for *biometric template protection*(BTP) technique.

Specifically, BTP aims to achieve three goals: *irreversibility*, *renewability* and *unlinkability* with considerable recognition performance:

- **Irreversibility**: Prevents the reconstruction of the original biometrics, ensuring one-time usage security.
- **Renewability/Reusability**: Allows secure reissue of protected templates even after leakage of old protected templates, enabling multiple uses.
- **Unlinkability**: Ensures protected templates from same individual cannot be linked to a single identity.

Notably, unlinkability is the most robust property and considerably more challenging to achieve, compared to the irreversibility and the renewability.

Fuzzy-based scheme is a promising technique to implement BTP. Fuzzy-based schemes include fuzzy extractor [13], fuzzy vault [22] and fuzzy commitment [23]. They commonly consist of two core components:

- **Information Reconciliation**: Maps similar readings to an identical value, typically applied by secure sketch based on an error-correcting code (ECC);
- **Privacy Amplification**: Converts the uniform value to an uniformly random secret string, typically accomplished by extractors or cryptographic hash function.

While fuzzy-based scheme has been successfully applied to binary space with binary error-correcting codes[3, 31] and real-valued space $\mathbb{R}^n$ with lattice code[19, 24], hypersphere-based implementations without directly transforming to the binary space remained elusive until **IronMask** [26]. In [26], Kim et al. devise an error correcting code on hypersphere to build a secure sketch and in turn a fuzzy commitment scheme, named **IronMask**. They apply IronMask to protect facial template on **ArcFace** with template dimension as $n = 512$, recommending the error correcting parameter $\alpha = 16$. The combination achieves a true accept rate(TAR) of 99.79% at a false accept rate(FAR) of 0.0005% and providing at least 115-bit security against known attacks. They claimed that their scheme satisfies irreversibility, renewability and unlinkability. To the best of our knowledge, it's the best BTP scheme to provide high security while preserving facial recognition performance without other secrets.

1.1 Our Contributions.

Our key contributions are three:

1. Novel Attack Methodology: We propose a new method named as **Probabilistic Linear Regression Attack** that utilizes the linearity in **IronMask**'s error-correcting code to compromise the renewability of **IronMask**. Let n denote the dimension of the output template and α denote the error correcting parameter, the algorithm exhibits $O(n^p e^{c\alpha})$ complexity, where p depends on the theoretical complexity of underlying linear regression solver and c scales with the number of obtained protected templates.

*Corresponding author.

Proceedings on Privacy Enhancing Technologies 2025(4), 728–744

https://doi.org/10.56553/popets-2025-0154

2. Comprehensive Validation: We apply the probabilistic linear regression attack on **IronMask** applied to protect **ArcFace** and carry out experiments that well verify the validity of our attacks. The experiment is carried out on a single laptop with Intel Core i7-12700H running at 2.30 GHz and 64 GB RAM. The experimental results with different linear regression solvers, SVD (Singular Value Decomposition) solver and LSA (Locality Sensitive Algorithm) solver, and different settings demonstrates:

- SVD solver(No Noise): 5.3 days with $n - 1 = 511$ protected templates and 621 days with 2 protected templates.
- LSA solver(No Noise): 4.8 days with 3 protected templates and 7.1 days with 281 protected templates.
- Noise resilience: at most 50 years recovery time with 3 protected templates at noise less than 36° when deploying LSA-based linear solver.
- Real-World Dataset performance: around 35.6 years(Color-FERET) and around 1.47 years (FEI).

Our experimental results show that the attack maintains effectiveness in noisy environments and could be applied in real-world scenarios.

3. Defense Strategies: We propose two plausible defence strategies, Adding Extra Noise and Salting on strengthening **IronMask**, in order to mitigate our attacks. When combined, these strategies could boost security to 60-bit. While the costs are lower recovery success probabilities(95%), slower authentication(3 seconds) and significant storage requirement(~80GB).

2 Related Works

2.1 Face Recognition on Hypersphere

Modern face recognition based on deep neural networks generally operate through two fundamental phases: (1) generating an embedded facial template from input images, and (2) computing similarity scores between templates for identity matching.

Current research has demonstrated that angular margin-based losses significantly outperform traditional contrastive loss[10] and triplet loss[45] on large-scale dataset, such as SphereFace[32], **ArcFace**[12] and MagFace[34]. These face recognition frameworks constrain the face templates to a unit-hypersphere, with the cosine similarity score between $\mathbf{w}_1$ and $\mathbf{w}_2$ calculated as $\text{Score}(\mathbf{w}_1, \mathbf{w}_2) = \arccos \frac{\mathbf{w}_1 \cdot \mathbf{w}_2}{|\mathbf{w}_1||\mathbf{w}_2|}$.

2.2 Fuzzy-based BTP on Face Recognition

For neural network based face recognition systems, numerous protection techniques utilizing error-correcting code(ECC) or secure sketch have emerged. Existing approaches primarily process binary representation templates and can be categorized into:

- **Direct learning method**: Establish end-to-end mappings from face images to binary codes [39, 47].
- **Transformation method**: Convert real-valued templates to binary representations via recording auxiliary information [1, 35].

However, these approaches both typically incur significant performance degradation due to the loss of discriminatory information during the translation from real space to binary space.

For Transformation method, recent researches are as follows. Rathgeb et al. employ Linearly Seperable SubCode(LSSC) to extract binary representation of face template and apply a fuzzy vault scheme to protect it [43]. They claim around 32 bits false accept security analysed in Color-FERET dataset[41]. Jiang et al. in [20] also transform the face template to binary code but using computational secure sketch, which is based on DMSP assumption [16], to implement face-based authentication scheme and achieve considerable performance. However the assumption is new and requires further security validation. And the scheme lacks the security analysis of renewability and unlinkability.

IronMask [26] represents a breakthrough as the first fuzzy commitment scheme operating directly on hypersphere space, achieving better performance than other protection techniques based on ECC, as maintaining 99.79% TAR at 0.0005% FAR when protecting **ArcFace**. They demonstrate that their scheme satisfies irreversiblity, renewability and unlinkability to known attacks in their parameter settings when protecting **ArcFace**. Under particular settings, they claim that it can provide at least 115-bit security against known attacks($n = 512$, $\alpha = 16$).

2.3 Attack on Fuzzy-based Scheme

For secure sketch and fuzzy extractor on binary space, several analyses and attacks have targeted core security properties such as irreversibility, reusability and unlinkability. [4, 46] find that the original template can be recovered when getting multiple sketches from same template if the underlying error-correcting codes are different or biased. [46] finds an attack that can break the unlinkability of secure sketch. However, their attacks and analysis are focusing on the secure sketch within binary space and are not suitable for targeting the hypersphere space. To date, no effective attacks have been demonstrated against the reusability and unlinkability of the secure sketch on the hypersphere.

3 Revisit HyperSphere Error Correcting Code and Secure Sketch

3.1 Notations

We denote the set $\{1, 2, \cdots, \rho\}$ as $[\rho]$. Denote general space, particularly biometric template spaces, as calligraphic letters ,such as $\mathcal{M}$, with the following particular space: $\mathbb{R}^n$ as n-dimension Euclidean space, $S^{n-1} \triangleq \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 = 1\}$ as unit hypersphere space in $\mathbb{R}^n$ and $O(n)$ as orthogonal group in $\mathbb{R}^n$(set of $n \times n$ orthogonal matrices).

The vector in $\mathbb{R}^n$ is denoted by bold small letter like $\mathbf{v}$, while the matrix is denoted by bold large letter like $\mathbf{M}$. Angular distance metric S^{n-1} is defined as $\mathbf{v}, \mathbf{w}$ is defined as

$$Angle(\mathbf{v}, \mathbf{w}) \triangleq \arccos\left(\frac{\langle \mathbf{v}, \mathbf{w} \rangle}{\|\mathbf{v}\|_2 \|\mathbf{w}\|_2}\right)$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product and $\| \cdot \|_2$ the Euclidean norm.

In this paper, we concentrate on the metric space S^{n-1} with Angle distance, as it is the embedding space of face template space $\mathcal{M}$ of **ArcFace**.

3.2 HyperSphere Error Correcting Code

Here we recall the definition of the hypersphere error-correcting code (HyperSphere-ECC), which serves as the foundation for constructing hypersphere secure sketch (HyperSphere-SS).

Definition 3.1 (HyperSphere-ECC [26]). A set of codewords $C \subset S^{n-1}$ is called HyperSphere-ECC if there exists θ such that:

(1) (Discriminative) $\forall \mathbf{c}_1, \mathbf{c}_2 \in C$, $Angle(\mathbf{c}_1, \mathbf{c}_2) > \theta$;
(2) (Efficiently Decodable) There exists an efficient algorithm **Decode**, such that $\forall \mathbf{c} \in C, \forall \mathbf{a} \in S^{n-1}$, if $Angle(\mathbf{a}, \mathbf{c}) < \frac{\theta}{2}$, $\mathbf{Decode}(\mathbf{a}) = c$.

where we call θ as the design distance.

Kim et al. [26] devised a family of HyperSphere-ECC that can be efficiently sampled and decoded.

Definition 3.2. [26] For any positive integer α, C_α is a set of codewords which have exactly α non-zero entries. Each non-zero entries are either $-\frac{1}{\sqrt{\alpha}}$ or $\frac{1}{\sqrt{\alpha}}$.

Theorem 3.3 (Design Distance[26]). *The design distance θ for C_α is* $\frac{1}{2}\arccos(1-\frac{1}{\alpha})$.

In real world, even for $\mathbf{c} \in C, \mathbf{a} \in S^{n-1}, Angle(\mathbf{a}, \mathbf{c}) > \frac{\theta}{2}$, there are chances that $\mathbf{Decode}(\mathbf{a}) = \mathbf{c}$, as the theoretical design distance θ provides a conservative guarantee.

Algorithm 1: Sample and Decode Algorithms for HyperSphere-ECC C_α in Definition 3.2

Function Sample$(n, \alpha) \to \mathbf{c}$
- **Data:** dimension n, error parameter α
- **Result:** $\mathbf{c} \in C_\alpha$
- Random choose α distinct positions $j_1, \ldots, j_\alpha$;
- $\mathbf{c} \leftarrow (0, \ldots, 0)_n$;
- **for** $i \in \{j_1, \ldots, j_\alpha\}$ **do**
  - $c_i \xleftarrow{\$} \{-\frac{1}{\sqrt{\alpha}}, \frac{1}{\sqrt{\alpha}}\}$;
- Output $\mathbf{c}$;

Function Decode$(u, \alpha) \to \mathbf{c}$
- **Data:** $\mathbf{u} \in S^{n-1}$, error parameter α
- **Result:** $\mathbf{c} \in C_\alpha$
- Find the best α positions $J = \{j_1, \ldots, j_\alpha\}$ such that $\forall j \in J, \forall k \in [n]/J |u_j| \geq |u_k|$;
- $\mathbf{c} \leftarrow (0, \ldots, 0)_n$;
- **for** $i \in J = \{j_1, \ldots, j_\alpha\}$ **do**
  - $c_i \leftarrow \frac{u_i}{|u_i|\sqrt{\alpha}}$;
- Output $\mathbf{c}$;

3.3 HyperSphere Secure Sketch and IronMask Scheme

The secure sketch premitive, first proposed in [13] by Dodis et al., enables precise recovery of $\mathbf{w}$ from any $\mathbf{w}'$ close to $\mathbf{w}$ with public information while not revealing too much information of $\mathbf{w}$. It has been a basic component to construct fuzzy extractor [13] and fuzzy commitment [23]. Here we recall the definition of the secure sketch.

Definition 3.4 (Secure Sketch[13]). An $(\mathcal{M}, t)$ secure sketch consists of a pair of algorithms $(\mathbf{SS}, \mathbf{Rec})$.

- The sketching algorithm **SS** takes input $\mathbf{w} \in \mathcal{M}$, outputs sketch s as public information.
- The recovery algorithm **Rec** takes inputs $\mathbf{w}' \in \mathcal{M}$ and sketch s, outputs w''.

It satisfies the following properties:

- Correctness: if $dis(\mathbf{w}, \mathbf{w}') < t$, then $\mathbf{Rec}(\mathbf{w}', \mathbf{SS}(\mathbf{w})) = \mathbf{w}$;
- Security: For security parameter λ, either:
  - Information-theoretic: $\max_{\mathbf{w}} \Pr[\mathbf{w}|s = \mathbf{SS}(\mathbf{w})] < \frac{1}{2^\lambda}$.
  - Computational: $\forall$ Probabilistic Polynomial Time (PPT) adversary $\mathcal{A}$, $\Pr[\mathcal{A}(s = \mathbf{SS}(\mathbf{w})) = \mathbf{w}] < \frac{1}{2^\lambda}$.

For space $\mathcal{F}^n$ with hamming distance, Dodis et al. proposes a general construction of secure sketch based on error-correcting code [13]. They also devise a general construction on the transitive space $\mathcal{M}$, i.e. for any $a, b \in \mathcal{M}$, there exists an isometry transformation π satisfying $b = \pi(a)$. Since hypersphere space is also transitive with orthogonal matrices, we can build a hypersphere secure sketch based on HyperSphere-ECC similar to error-correcting code.

Definition 3.5 (HyperSphere Secure Sketch). For a HyperSphere-ECC C with decode algorithm **Decode** and design distance θ, the hypersphere secure sketch can be constructed as below:

- Sketching algorithm **SS**: on input $\mathbf{w} \in S^{n-1}$, $\mathbf{c} \xleftarrow{\$} C$, randomly generate an orthogonal matrix $\mathbf{M}$ that satisfies $\mathbf{c} = \mathbf{Mw}$, output $\mathbf{M}$ as sketch;
- Recovery algorithm **Rec**: on inputs $\mathbf{w}' \in S^{n-1}$ and orthogonal matrix $\mathbf{M}$, compute $\mathbf{v} \leftarrow \mathbf{Mw}'$, $\mathbf{c}' \leftarrow \mathbf{Decode}(\mathbf{v})$, output $\mathbf{w}'' \leftarrow \mathbf{M}^{-1}\mathbf{c}'$.

It satisfies the following properties:

- Correctness: If $Angle(\mathbf{w}, \mathbf{w}') < \frac{\theta}{2}$, $Angle(\mathbf{Mw}, \mathbf{Mw}') < \frac{\theta}{2}$. Based on the correctness of **Decode** algorithm, as $\mathbf{Mw} = \mathbf{c}$, $\mathbf{c}' = \mathbf{Decode}(\mathbf{Mw}') = \mathbf{c}$. Thus $\mathbf{w}'' = \mathbf{M}^{-1}\mathbf{c}' = \mathbf{M}^{-1}\mathbf{c} = \mathbf{w}$.
- Security: Since $\mathbf{M}$ is randomized, if $\mathbf{w}$ samples from uniform distribution on S^{n-1}, $\{\mathbf{M}^{-1}\mathbf{c}|\forall \mathbf{c} \in C\}$ is set of all possible inputs with equal probability of **SS**. The probability for adversary guessing correct answer at one attempt is $\frac{1}{|C|}$.

Kim et al. [26] implement an algorithm named **hidden matrix rotation** to generate the random orthogonal matrix $\mathbf{M}$ with constraint $\mathbf{c} = \mathbf{Mw}$.

3.3.1 Tradeoff of Correctness and Security. For secure sketch based on ECC, the error correcting capability of ECC is an important parameter to control the usability and the security of the whole algorithm. To achieve high security, the error correcting capability would be sacrificed so as usability. In [26], they choose $\alpha = 16$ and $n = 512$ to achieve $|C| = \binom{n}{\alpha} \cdot 2^\alpha \approx 2^{115}$ security with average degrades 0.18% of true accept rate(TAR) at the same false accept rate(FAR) compared to facial recognition system without protection.

3.3.2 Usage in Fuzzy Commitment and IronMask Scheme. The secure sketch primitive can be directly combined with cryptographic hashing to construct a fuzzy commitment scheme [23]. The authentication protocol operates as follows:

- **Enrollment Phase**:

(1) Generate protected template: $\mathbf{SS}(\mathbf{w}) \rightarrow (c, p)$;
(2) Compute hash commitment: $H(\mathbf{c})$;
(3) Server stores: $(p, H(\mathbf{c}))$.

- **Authentication Phase**:

(1) Client recovers: $\mathbf{c}' = \mathbf{Rec}(\mathbf{w}', p)$
(2) Server verifies: $H(\mathbf{c}') \stackrel{?}{=} H(\mathbf{c})$

IronMask [26] implements this paradigm by replacing the secure sketch scheme with HyperSphere-SS based on particular implementation of HyperSphere-ECC in Definition 3.2.

Hash function is computationally secure if the probability of correctly guessing codeword $\mathbf{c}$ in one trial is small. However, even for high probability of guessing secret codeword $\mathbf{c}$(e.g. 2^{-40} if $|C| = 2^{40}$), take advantage of slow hashes, such as PBKDF2 [36], bcrypt [42] and scrypt [40], it's still inapplicable to implement exhausted searching attack, offering realistic security. Thus, we can strategically relax the computational security requirements through careful balancing of security and efficiency considerations.

3.4 Threat Model

Here we define a game version of multiple usage security (reusability) of secure sketch. Note that if $t = 0$, the multiple usage security degenerates to irreversibility.

Definition 3.6. Let $SS = (\mathbf{SS}, \mathbf{Rec})$ be secure sketch's two algorithms. The experiment $\mathrm{SSMUL}_{\mathcal{A},\theta,t}(n)$ is defined as follows:

(1) The challenger C chooses a biometric resource $\mathcal{W}$, samples $\mathbf{w} \in \mathcal{W}$ and sends $\mathbf{SS}(\mathbf{w})$ to adversary $\mathcal{A}$;
(2) $\mathcal{A}$ asks $q \leq t$ queries to challenger C. C samples $\{\mathbf{w}_i \in \mathcal{W}, i = 1, \ldots, q\}$ with constraints that $\forall i, j \in \{1, \ldots, q\}$, $\mathbf{w}^T\mathbf{w}_i \geq \cos\theta \wedge \mathbf{w}_i^T\mathbf{w}_j \geq \cos\theta$. Calculate the response set $Q = \{\mathbf{SS}(\mathbf{w}_i), i = 1, \ldots, q\}$ and send Q to $\mathcal{A}$;
(3) $\mathcal{A}$ outputs $\mathbf{w}'$. If $\mathbf{w}' = \mathbf{w}$, outputs 1, else outputs 0.

The secure sketch is secure with $(t+1)$ multiple usage if existing negligible function *negl* such that $\Pr[\mathrm{SSMUL}_{\mathcal{A},\theta,t}(n) = 1] < negl(n)$ for all **PPT** adversaries $\mathcal{A}$.

The definition is very similar to the reusability of fuzzy extractor. However, our definition allows the attacker to control the distance of each sampled templates from same source while reusable fuzzy extractor does not [2, 49] or assumes too powerful attacker with ability to totally control shift distance between each templates in binary space [4]. We argue that our definition can more accurately capture the attacker's power to recover template in real scenarios. As in real world, the attacker is more probable to get multiple sketches from different servers but can not accurately control the shift distance enrolled each time. But it might get a vague quality report of each enrolled sketch and select the sketches that have similar qualities thus bounding the angle distance of pairs of corresponding unprotected templates, consistent with our security model.

For more general protect algorithms, we may loose the target of attacker to only get a template close to original template, as in definition 3.7, where the attacker's target is to retrieve template $\mathbf{w}'$ satisfying $\mathbf{w}'^T\mathbf{w} \geq \cos\theta'$. However, for hypersphere secure sketch in Definition 3.5, the two definitions are somewhat equivalent (see in Theorem C.1 and Appendix C) if θ' is less or equal to the secure sketch's criterion to judge whether two templates are from the same individual, i.e. the recovery algorithm could retrieve the original template. Because intuitively the attacker in Definition 3.7 could use secure sketch's recovery algorithm to retrieve the original template with non-negligible probability, thus threatening the security in Definition 3.6. And in our scenario, the attacker can only successfully authenticate to the fuzzy commitment scheme if it retrieves the exact original template $\mathbf{w}$ and computes $H(\mathbf{c})$ where $\mathbf{c} = \mathbf{Mw}$ and $\mathbf{M} = \mathbf{SS}(\mathbf{w})$. Thus, we take Definition 3.6 as our threat model.

Definition 3.7 (General Biometric Template Protection Algorithm's Multiple Usage Security). Let P be biometric protection scheme S's generation algorithm. The experiment $\mathrm{SSMUL}_{\mathcal{A},\theta,t,\theta'}(n)$ is defined as follows:

(1) The challenger C chooses a biometric resource $\mathcal{W}$, samples $\mathbf{w} \in \mathcal{W}$ and sends $\mathbf{P}(\mathbf{w})$ to adversary $\mathcal{A}$;
(2) $\mathcal{A}$ asks $q \leq t$ queries to challenger C. C samples $\{\mathbf{w}_i \in \mathcal{W}, i = 1, \ldots, q\}$ with constraints that $\forall i, j \in \{1, \ldots, q\}$, $\mathbf{w}^T\mathbf{w}_i \geq \cos\theta \wedge \mathbf{w}_i^T\mathbf{w}_j \geq \cos\theta$. Calculate the response set $Q = \{\mathbf{P}(\mathbf{w}_i), i = 1, \ldots, q\}$ and send Q to $\mathcal{A}$;
(3) $\mathcal{A}$ outputs $\mathbf{w}'$. If $\mathbf{w}'^T\mathbf{w} \geq \cos\theta'$, outputs 1, else outputs 0.

The biometric protection scheme $\mathbf{S}$ is secure with $(t+1)$ multiple usage if existing negligible function *negl* such that for all **PPT** adversaries $\mathcal{A}$, $\Pr[\mathrm{SSMUL}_{\mathcal{A},\theta,t,\theta'}(n) = 1] < negl(n)$.

4 Probabilistic Linear Regression Attack

4.1 Core Idea

Hypersphere secure sketch is secure from the information theoretical view in one-time usage. While if the same template $\mathbf{w}$ is sketched multiple times, the sketches can determine the original $\mathbf{w}$.

For example, if $\mathbf{w}$ is sketched twice as $\mathbf{w}_1 = \mathbf{w}$ and $\mathbf{w}_2$ where $\epsilon = \mathbf{w}_2 - \mathbf{w}_1$ is the noise introduced during the second sampling process, assume the sketches are $\mathbf{M}_1, \mathbf{M}_2$ and corresponding codewords are $\mathbf{c}_1, \mathbf{c}_2$. The codeword pair $(\mathbf{c}_1, \mathbf{c}_2)$ satisfies that $\mathbf{c}_2 = \mathbf{M}_2\mathbf{M}_1^{-1}\mathbf{c}_1 + \epsilon'$ where $\epsilon' = \mathbf{M}_2\epsilon$. Since $\mathbf{M}_1, \mathbf{M}_2$ are random orthogonal matrix, $\mathbf{M} = \mathbf{M}_2\mathbf{M}_1^{-1}$ can be seen as a random orthogonal matrix with only one constraint that maps $\mathbf{c}_1$ to $\mathbf{c}_2 - \epsilon'$. Sparsity of the codewords implies there are few other pairs of $(\mathbf{c}_1', \mathbf{c}_2')$ satisfying $\mathbf{c}_2' = \mathbf{Mc}_1' + \epsilon_r$ where ϵ_r is random small noise, so as corresponding template $\mathbf{w}$, otherwise $\mathbf{M}$ should have other constraints and might even leak whole information of original template if $\forall \mathbf{c} \in C_\alpha, \mathbf{Mc} \in C_\alpha$(see in Section 6.1). By exhaustive searching in the space of codewords, the pair $(\mathbf{c}_1', \mathbf{c}_2')$ can be determined so as the noise ϵ' and the original template $\mathbf{w}$ can be recovered as $\mathbf{M}_1^{-1}\mathbf{c}_1'$. Even if the structure of HyperSphere-ECC can be used, it's possible to downgrade the computation complexity of recovering $\mathbf{w}$.

The codewords used in **IronMask** do have special structure. Based on the HyperSphere-ECC construction employed by **IronMask** in Definition 3.2, we have designed distance $\theta = \frac{1}{2}\arccos(1 - \frac{1}{\alpha})$. For satisfactory accuracy, the codeword $\mathbf{c}$ should utilize a small value of α(specifically, $n = 512, \alpha = 16$ are chosen). Therefore, the codewords contain a preponderance of $(n - \alpha)$ zeros. From another view, given that matrix $\mathbf{M} = \mathbf{SS}(\mathbf{w})$ and $\mathbf{m}_i$ be ith row vector of $\mathbf{M}$, we have

$$\Pr[\mathbf{m}_i \cdot \mathbf{w} = 0 | \mathbf{m}_i \text{ is the ith row vector of } \mathbf{M}, i \stackrel{\$}{\leftarrow} [n]] = \frac{n - \alpha}{n} \quad (1)$$

In the realm of linear algebra, determining the vector $\mathbf{w} \in S^{n-1}$ necessitates a minimum of $(n-1)$ linear equations. Each sketch matrix $\mathbf{M}$ derived from $\mathbf{SS}(\mathbf{w}+\epsilon)(\epsilon$ is small noise) offers a $\frac{n-\alpha}{n}$ probability of correctly yielding a linear equation of the form $\mathbf{w}^T\mathbf{v}' \approx 0$. Therefore, if we possess $(n-1)$ sketches and randomly select a row vector from each, the probability of obtaining $(n-1)$ correct linear equations amounts to $(\frac{n-\alpha}{n})^{n-1}$, which approximates to e^{α} when $\alpha \ll n$. These equations are highly probably linear independent as they can be seen as randomly and independently selected from $\{\mathbf{w}' + \epsilon | \mathbf{w}'^T\mathbf{w} = \mathbf{0}, \epsilon$ is random noise$\}$. Assume we obtain $(n-1)$ correct linear equations, utilizing singular value decomposition(SVD), we can then deduce either the original vector $\mathbf{w}$(without noise) or a closely related vector $\mathbf{w}'$(with noise). Additionally, by utilizing the recovery algorithm of hypersphere secure sketch, we can reconstruct the original vector even when provided with a noisy solution $\mathbf{w}'$.

Furthermore, through fully exploiting HyperSphere-ECC inherent structure, we could reduce the number of required linear equations, increasing the probability of obtaining correct linear equations at one try. Assuming that we possess n sketches relating to $\mathbf{w}$, denoted as $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n$ with corresponding codewords $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n$. We can deduce the equations

$$c_i = \mathbf{M}_i\mathbf{M}_1^{-1}c_1 + \epsilon_i', \forall 2 \le i \le n, \text{where } \mathbf{w}_n = \mathbf{w}_1 + \mathbf{M}_i\epsilon_i' \quad (2)$$

Defining $\mathbf{M}_i' = \mathbf{M}_i\mathbf{M}_1^{-1}$, we can interpret $\mathbf{M}_i'$ as sketches of $\mathbf{c}_1$. This allows us to solve for $\mathbf{c}_1$ using $(n-1)$ linear equations. However, given that the entries of $\mathbf{c}_1 \in C_\alpha$ predominantly consist of zeroes and that the non-zero entries possess uniform norms, the task of determining $\mathbf{c}_1$ based on k linear equations can be seen as the **Subset Sum Problem** or the **Sparse Linear Regression Problem**. Numerous algorithms exist for tackling such problems, such as [11, 30, 44] for the **Subset Sum Problem** and [5, 6, 9, 14, 17, 18] for the **Sparse Linear Regression Problem**. We choose to use the Local Search Algorithm(LSA) introduced by Gamarnik and Zadik in [18] due to its effectiveness and simplicity of implementation. And other algorithms are more applicable when the coefficients of the linear equations are independent of the input vector($\mathbf{c}_1$), which does not align with our specific problem where the equation's value is zero.

4.2 Details of Probabilistic Linear Regression Attack

The attack comprises three main components: the Linear Equation Sampler, the Linear Regression Solver, and the Threshold Determinant .

Initially, the linear equation sampler receives t sketches, denoted as $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_t$, and sample rows from them to construct a single matrix $\mathbf{M}$. Subsequently, the linear regression solver processes this matrix $\mathbf{M}$ and strives to generate a solution vector $\mathbf{w}'$ satisfying $||\mathbf{M}\mathbf{w}'|| \approx 0$. Finally, the threshold determinant utilizes $\mathbf{w}'$ to recover candidate template $\mathbf{w}_r$ through recovery algorithm of secure sketch. This component then determines whether the recovered template is correct based on the predefined threshold θ_t and the angle between $\mathbf{w}_r$ and $\mathbf{w}_{r'}$, which is output of recovery algorithm with inputs of $\mathbf{w}_r$ and another sketch.

Algorithm 2: Linear Equation Sampler

Data: Matrices $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_t \in \mathbf{SS}(\cdot)$, type = "SVD" or "LSA", k
Result: $\mathbf{M}$
if *type = "SVD"* **then**
  $t' \leftarrow t$
  for $i = 1, \ldots, t'$ **do**
    $\mathbf{M}_i' \leftarrow \mathbf{M}_i$
else if *type="LSA"* **then**
  $t' \leftarrow t-1$
  for $i = 1, \ldots, t'$ **do**
    $\mathbf{M}_i' \leftarrow \mathbf{M}_{i+1}\mathbf{M}_1^{-1}$
$l \leftarrow \lfloor k/t' \rfloor$
for $i = 1, \ldots, t'$ **do**
  $\{\mathbf{v}_{l(i-1)+j}^T | 1 \le j \le l\} \leftarrow$ Random select different l row vectors of $\mathbf{M}_i'$
for $i = 1, \ldots, k - l * t'$ **do**
  $\{\mathbf{v}_{l*t'+i}^T\} \leftarrow$ Random select row vector different from already sampled vectors of $\mathbf{M}_i'$
$\mathbf{M} \leftarrow$ vertical stack of $\mathbf{v}_1^T, \mathbf{v}_2^T, \ldots, \mathbf{v}_k^T$

Algorithm 3: Linear Regression Solver based on SVD

Data: Matrix $\mathbf{M}$ with size $k * n$ where $k \ge n-1$
Result: $\mathbf{w} = \text{argmin}_{\mathbf{w}} ||\mathbf{M}\mathbf{w}||$ where $\mathbf{w} \in S^{n-1}$
$\mathbf{w} \leftarrow$ the eigenvector of matrix $\mathbf{M}$ with smallest eigenvalue

4.2.1 Linear Equation Sampler. The Linear Equation Sampler receives the sketches $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_t$ derived from template $\mathbf{w}$ as input. Depending on the linear regression solver's chosen algorithm, the sampler selects row vectors from these sketches differently. If the solver employs the SVD algorithm, the sampler randomly picks k row vectors from the sketches. If solver uses LSA, the sampler first computes $t-1$ matrices $\mathbf{M}_i' = \mathbf{M}_{i+1}\mathbf{M}_1^{-1} \forall 2 \le i \le t$ and then randomly select k row vectors from these matrices. Subsequently, the sampler vertically stacks the chosen vectors to form the matrix $\mathbf{M}$ which is the input of the linear regression solver. The sampler's duty is to maximum the likelihood that $||\mathbf{M}\mathbf{w}|| \approx 0$(or $||\mathbf{M}\mathbf{M}_1\mathbf{w}|| \approx 0$). We deem the matrix $\mathbf{M}$ as "correct" if for each selected row vector, the entry in corresponding mapped codeword is 0, where $\mathbf{w}$ represents original input template. "Correct" matrix ensures that $||\mathbf{M}\mathbf{w}|| \approx 0$.

Definition 4.1. In Algorithm 2, assume row vector $\mathbf{v}_i$ is sampled in m'th row of matrix $\mathbf{M}_j'$ and the mapped codeword of $\mathbf{M}_j'$ is $\mathbf{c}$, i.e. $\mathbf{M}_j'\mathbf{w} = \mathbf{c}$ if type = "SVD" where $\mathbf{M}_j = \mathbf{SS}(\mathbf{w})$ or $\mathbf{M}_{j+1}\mathbf{w} = \mathbf{c}$ if type = "LSA" where $\mathbf{M}_{j+1} = \mathbf{SS}(\mathbf{w})$. We say the row vector $\mathbf{v}_i$ sampled by linear equation sampler is "correct" if and only if the m'th entry of $\mathbf{c}$ is 0, i.e. $\mathbf{c}_m = 0$. We say the sampled matrix $\mathbf{M}$ is "correct" if and only if all row vectors sampled are "correct". Otherwise, $\mathbf{M}$ is "incorrect".

4.2.2 Linear Regression Solver. The linear regression solver takes the output matrix $\mathbf{M}$ as input, solves the following optimization

Algorithm 4: Linear Regression Solver based on Local Search Algorithm(LSA)

Data: Matrix $\mathbf{M}$ with size $k * n$, error correcting code parameter α, hyper-parameter d, t_{th}
Result: $\mathbf{c}$ or $\perp$
$t \leftarrow 0$
repeat
  $\mathbf{c} \leftarrow$ Random select codeword in C_α
  $t \leftarrow t + 1$
  repeat
    $norm_{pre} \leftarrow ||\mathbf{Mc}||$
    $norm_{new} \leftarrow ||\mathbf{Mc}||$
    for $\mathbf{c}' \in C_\alpha$ *where* $||\mathbf{c}' - \mathbf{c}|| = \frac{\sqrt{2}}{\sqrt{\alpha}}$ **do**
      $norm_{tmp} \leftarrow ||\mathbf{Mc}'||$
      if $norm_{tmp} < norm_{new}$ **then**
        $norm_{new} \leftarrow norm_{tmp}$, $\mathbf{c}_{new} \leftarrow \mathbf{c}'$
    $\mathbf{c} \leftarrow \mathbf{c}_{new}$
  until $norm_{new} = norm_{pre}$
until $||\mathbf{Mc}|| \le d$ *or* $t > t_{th}$
if $||\mathbf{Mc}|| \le d$ **then**
  Output $\mathbf{c}$
else
  Output $\perp$

problem:

$$\text{argmin}_{\mathbf{w}} ||\mathbf{Mw}||, \mathbf{w} \in S^{n-1} \tag{3}$$

Two algorithms are employed : SVD(Singular Vector Decomposition) and LSA(Local Search Algorithm).

SVD can be used to find the null space of a matrix by identifying the singular vectors associated with zero (or near-zero) singular values, which represent vectors transformed close to zero. For SVD-based solver, a minimum of $(n-1)$ linear equations are required, and the output $\mathbf{w}'$ is a candidate solution for the original template. If only two sketches, $\mathbf{M}_1, \mathbf{M}_2$, are available in the step of linear equation sampler, an approximate solution of equation 3 can be obtained by solving a smaller matrix. When given 2 sketches, the task of linear equation sampler is equivalent to guessing $\frac{k}{2}$ zero entries in each corresponding codewords $\mathbf{c}_1 = (c_1^1, c_2^1, \ldots, c_n^1)$, $\mathbf{c}_2 = (c_1^2, c_2^2, \ldots, c_n^2)$, thus total k zero entries. Assuming U_1 and U_2 are the guessed set of zero-value entries in $\mathbf{c}_1$ and $\mathbf{c}_2$ respectively, and given that $\mathbf{c}_2 = \mathbf{M}_2\mathbf{M}_1^{-1}\mathbf{c}_1$, we can formulate a set of linear equations. These equations relate the non-zero entries of $\mathbf{c}_1$ to the zero entries of $\mathbf{c}_2$ through the matrix $\mathbf{M}_2\mathbf{M}_1^{-1}$. Therefore, assume $\mathbf{M}_2\mathbf{M}_1^{-1} = (m_{ij})$, we have

$$\sum_{j \notin U_1} m_{ij} c_j^1 = 0, \forall i \in U_2. \tag{4}$$

By applying SVD, we could find a solution $\mathbf{c}'$ that minimizes the squared error of these equations, subject to the constraint that the entries of $\mathbf{c}'$ indexed by U_1 are zero. The approximate solution for the original template is then given by $\mathbf{w}' = \mathbf{M}_1^{-1}\mathbf{c}'$. This approach reduces the matrix size from the original $k * n$ to $\frac{k}{2} * (n - \frac{k}{2})$ by at least factor 2 when $k \ge n - 1$.

For the LSA-based solver, the required number of linear equations exceeds $\alpha \log n$ [18]. The solution obtained is a codeword $\mathbf{c}' \in C_\alpha$, and the candidate template solution is derived as $\mathbf{M}_1^{-1}\mathbf{c}'$.

4.2.3 Threshold Determinant. The threshold determinant obtains the solution template vector $\mathbf{w}'$ from linear regression solver as an input and proceeds to attempt the recovery of the original template $\mathbf{w}$. Subsequently, it invokes the secure sketch's recovery algorithm utilizing $\mathbf{w}'$ and the sketch $\mathbf{M}_1$ to obtain candidate template $\mathbf{w}_{r_1}$. Then invoke the the secure sketch's recovery algorithm utilizing $\mathbf{w}_{r_1}$ and the sketch $\mathbf{M}_2$ to obtain another candidate template $\mathbf{w}_{r_2}$. Then the determinant calculates the angle θ' as $Angle(\mathbf{w}_{r_1}, \mathbf{w}_{r_2})$. If θ' surpasses a preset threshold θ_t, the determinant returns a false output, indicating to the linear equation sampler that a new matrix should be generated for the linear regression solver to process. Otherwise, it outputs $\mathbf{w}_{r_1}$ as the recovered solution template.

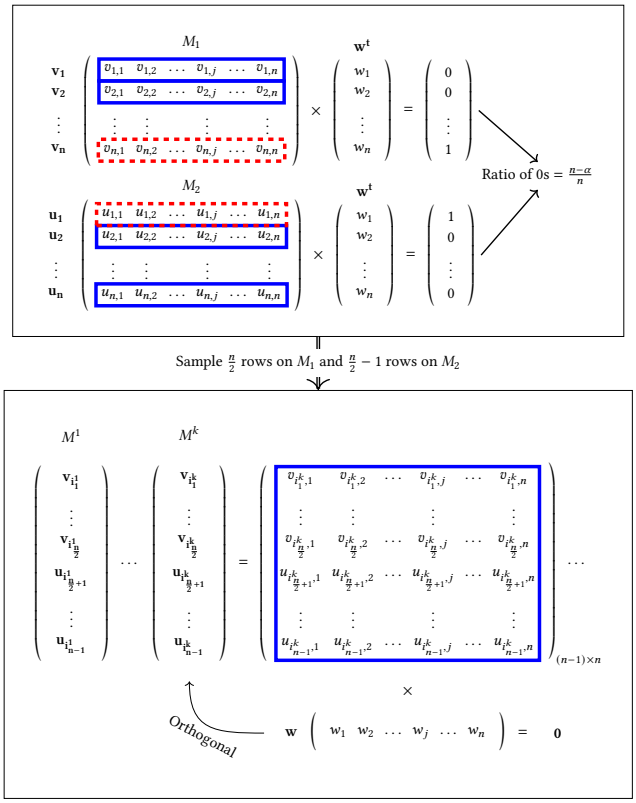

Figure 1: Overview of probabilistic linear regression attack based on SVD on two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$. The blue solid box indicates that the row vector is orthogonal to template $\mathbf{w}$ while the red dashed box is not. By randomly selecting $(n-1)$ row vectors, we finally get matrix $\mathbf{M}^k$ that $\mathbf{w}$ is in null space of $\mathbf{M}^k$. As $\mathbf{M}^k$ is full of rank, the only one null vector is parallel to $\mathbf{w}$.

4.3 Correctness and Complexity Analysis

In this section, we present an analysis of the correctness and computational complexity of the probabilistic linear regression algorithm,

specifically focusing on noiseless scenario. As for noisy environments, we demonstrate the practicality and efficiency of our algorithms through empirical experiments discussed in Section 5.

4.3.1 Correctness. The proof of the correctness of the probabilistic linear regression attack comprises three primary steps. First, we demonstrate that the inverse probability(p_s) of the output matrix $\mathbf{M}$ from the linear equation sampler is "correct" in Definition 4.1, which satisfies $\mathbf{Mw} = 0$(or $\mathbf{MM}_1\mathbf{w} = 0$), is equal to $2^{O(\alpha)}$(Theorem 4.2). Secondly, we establish that if the input matrix is "correct", the solution derived from linear regression solver is parallel to original template $\mathbf{w}$(Theorem 4.3). Lastly, we show that the threshold determinant effectively filters out solutions $\mathbf{w}'$ corresponding to "incorrect" sampled matrices as "incorrect" matrix's solution deviates the original template(Theorem 4.4).

Theorem 4.2. *When $\alpha \ll n$ and $k \leq n-1$, let the secure sketches derived from the same template $\mathbf{w}$ be denoted as $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_t\}$, where $k = l \times t'$ (when type="LSA", $t' = t-1$; when type="SVD, $t' = t$). If $t' \geq 2$, then*

$$p_s = \Pr[\mathbf{M} \textit{ is "correct"}] \approx 2^{-o(\alpha)}. \tag{5}$$

Proof. Since $k = l \times t'$, the number of distinct row vectors sampled from the same matrix is l. The associated probability that l sampled row vectors from a matrix are "correct" is $\frac{\binom{l}{n-\alpha}}{\binom{l}{n}} \geq (1 - \frac{\alpha}{n-l+1})^l$. Consequently, the probability p_s that all vectors are "correct", i.e., the matrix is "correct", is:

$$\begin{aligned}
\Pr[\mathbf{M} \textbf{ is "correct"}] &= \left(\frac{\binom{l}{n-\alpha}}{\binom{l}{n}}\right)^{t'} \geq \left(1 - \frac{\alpha}{n-l+1}\right)^{l \times t'} \\
&= \left(1 - \frac{\alpha}{n-l+1}\right)^{k} \\
&\geq \left(1 - \frac{\alpha}{n-l+1}\right)^{n-1} \qquad \triangleright\, k \leq n-1 \\
&\approx e^{-\frac{\alpha n}{n-l+1}} \qquad \triangleright\, \alpha << n \\
&\geq e^{-2\alpha} \qquad \triangleright\, t' \geq 2.
\end{aligned}$$

Thus, we finish the proof. □

Theorem 4.3. *When the matrix $\mathbf{M}$ is "correct" and $k \leq n-1$ satisfies the conditions(the SVD solver requires $k = n-1$, and the LSA solver requires $k \geq o(\alpha \log(n))$), the solution $\mathbf{w}'$ obtained by the linear regression solver satisfies $\mathbf{w}' = \pm\mathbf{w}$, where $\mathbf{w}$ is the original template.*

Proof. Leveraging the correctness of SVD algorithm, we have $\mathbf{M}(\mathbf{w} - \mathbf{w}') = 0$ if $\mathbf{M}$ is "correct". Due to the rank of $\mathbf{M}$ is $n-1$, we have $\mathbf{w} = \pm\mathbf{w}'$.

While for the LSA-based solver, based on Theorem 2.7 from [18], for a sufficiently large $k \geq \alpha \log(n)$ and small σ as n approaches infinity, we have the solution codeword $\mathbf{c}'$ and original codeword $\mathbf{c}$ satisfying $||\mathbf{c} - \mathbf{c}'|| \leq \sigma$ and $\mathbf{c}' = \pm\mathbf{c}$ for small enough σ(noting that $-\mathbf{c}$ is also a valid solution for $\text{argmin}_{\mathbf{x}} ||\mathbf{Mx}||$). This implies that $\mathbf{w}' = \mathbf{M}_1^{-1}\mathbf{c}' = \mathbf{M}_1^{-1}(\pm\mathbf{c}) = \pm\mathbf{w}$. □

Theorem 4.4. *Assume $\theta_t = 0$, and let $\mathbf{w}'$ be the solution obtained by the linear regression solver. The threshold determinant outputs $\mathbf{w}'' = \pm\mathbf{w}$ where $\mathbf{w}$ is the original template when $\mathbf{w}' = \pm\mathbf{w}$; otherwise, outputs $\perp$.*

Proof. The threshold determinant outputs $\mathbf{w}'' = \mathbf{w}_{r_1}$ if and only if there exists $\mathbf{w}_{r_1}$ such that $\mathbf{w}_{r_2} = \mathbf{Rec}(\mathbf{w}_{r_1}, \mathbf{M}_2)$, $\mathbf{w}_{r_1}^T \mathbf{w}_{r_2} \geq \cos\theta_t$, which is equal to $\mathbf{w}_{r_2} = \mathbf{w}_{r_1}$. Let $\mathbf{c}_1' = \mathbf{M}_1\mathbf{w}_{r_1}$ and $\mathbf{c}_2' = \mathbf{M}_1\mathbf{w}_{r_2}$. Then, we can deduce that the threshold determinant outputs $\mathbf{w}''$ if and only if there are codeword pair $(\mathbf{c}_1', \mathbf{c}_2')$ satisfies:

$$\mathbf{c}_1' = \mathbf{M}_1\mathbf{M}_2^{-1}\mathbf{c}_2'. \tag{6}$$

Let $\mathbf{c}_1 = \mathbf{M}_1\mathbf{w}$ and $\mathbf{c}_2 = \mathbf{M}_1\mathbf{w}$. Due to the randomness of $\mathbf{M}_1$ and $\mathbf{M}_2$, $(\pm\mathbf{c}_1, \pm\mathbf{c}_2)$ are the only solutions satisfying the Equation 6. Therefore, $\mathbf{w}_{r_1} = \pm\mathbf{w}$ if $\mathbf{w}' = \pm\mathbf{w}$. Otherwise, the output is $\perp$. □

Theorem 4.5. *When $\alpha \ll n$ and $k = l \times t' \leq n-1$ satisfies the conditions(the SVD solver requires $k = n-1$, and the LSA solver requires $k \geq o(\alpha \log(n))$). Set $\theta_t = 0$. The expectation of the number of sample matrices the probabilistic linear regression attack $\mathcal{A}$ needs to output $\mathbf{w}' = \pm\mathbf{w}$, denoted as $E[\#\text{ of } \mathcal{A} \text{ tries} | \mathbf{w}' = \pm\mathbf{w}]$, satisfies:*

$$E[\#\textit{ of } \mathcal{A} \textit{ tries} | \mathbf{w}' = \pm\mathbf{w}] \leq 2^{o(\alpha)}. \tag{7}$$

Proof. Due to Theorem 4.2, Theorem 4.3 and Theorem 4.4, the probability p_a that the attacker outputs correct answer in one attempt satisfies:

$$\begin{aligned}
p_a \geq\ & \Pr[\mathbf{M} \textbf{ is "correct"}] \times \\
& \Pr[\text{solver outputs } \mathbf{w}' = \pm\mathbf{w} | \mathbf{M} \textbf{ is "correct"}] \times \\
& \Pr[\text{determinant outputs } \mathbf{w}'' = \pm\mathbf{w} | \text{solver outputs } \mathbf{w}' = \pm\mathbf{w}] \\
& \geq e^{-2\alpha} \times 1 \times 1 = e^{-2\alpha}.
\end{aligned}$$

Thus, we have

$$E[\#\text{ of } \mathcal{A} \text{ tries} | \mathbf{w}' = \pm\mathbf{w}] = \frac{1}{p_a} \leq e^{-2\alpha} = 2^{o(\alpha)}.$$

This completes the proof. □

4.3.2 Complexity. In the context of the linear equation sampler, the inverse of the probability that the output matrix $\mathbf{M}$ is "correct", which satisfies the condition $\mathbf{Mw} = 0$(or $\mathbf{MM}_1\mathbf{w} = 0$), is given by $2^{O(\alpha)}$, particularly e^{α} when l equals to 1 and $e^{2\alpha}$ when l equals to $\frac{n}{2}$ where $k = n-1$ and $l = k/t'$. For the linear regression solver and threshold determinant components, the algorithms employed exhibit polynomial time complexity with respect to the matrix size n. Consequently, the overall time complexity of the entire algorithm can be expressed as $2^{O(\alpha)}n^b$, where b is a constant representing the degree of the polynomial time complexity.

When it comes to the SVD linear regression solver, the SVD algorithm exhibits time complexity of $O(n^3)$ and it operates on a sampled matrix of dimension $(n-1) \times n$. Considering the entire algorithm, the time complexity is $O(e^{\alpha}n^3)$ for handling $n-1$ sketches, and $O(e^{2\alpha}n^3)$ for handling 2 sketches.

Regarding the LSA-based linear regression solver, each iteration carries a time complexity of $O(n^2)$. The maximum iteration count is influenced by factors such as α, σ and n, at least α for random initial vector. Nevertheless, there is no explicit formula indicating the precise number of equations necessary to arrive at accurate solutions. Consequently, determining the overall algorithm's complexity based on the local search method remains elusive. However, through empirical observations in Section 5, we hypothesize that

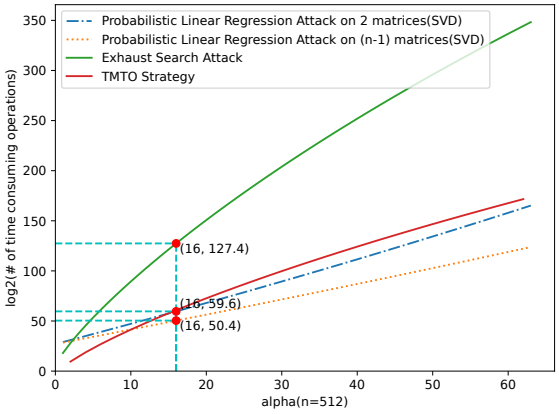

Figure 2: Log_2 of the number of most time-consuming operations(complexity) of each algorithm according to different α with $n = 512$. The complexity is $O(n^3e^{\alpha})$ for Algorithm 3 and $O(n^3e^{2\alpha})$ for optimized SVD-based solver given 2 sketch in Section 4.2.2. Here we take the constant number in the complexity of SVD algorithm as 1. However, for concrete algorithms, the constant number might be 8 or more. As this number is constant and small, we argue that it does not influence our conclusions.

the complexity of the local search-based algorithm is comparable to that of the SVD-based algorithm.

4.4 Comparison with TMTO Strategy

In [27], Kim et al. devise a time-memory-trade-off(TMTO) strategy to attack with two matrices, as solving $\mathbf{c}_2 = \mathbf{M}\mathbf{c}_1$ for $\mathbf{M} = \mathbf{M}_2\mathbf{M}_1^{-1}$. The core idea is that each codeword $\mathbf{c} \in C_\alpha$ can be seen as a combination of two codewords $\mathbf{c}_1'$ and $\mathbf{c}_1''$ from $C_{\frac{\alpha}{2}}$ with scalar $\frac{1}{\sqrt{2}}$ as $\mathbf{c}_2 = \frac{1}{\sqrt{2}}\left(\mathbf{M}\mathbf{c}_1' + \mathbf{M}\mathbf{c}_1''\right)$. We only need to compute the smaller set $\{\mathbf{M}\mathbf{c}'|c' \in C_{\frac{\alpha}{2}}\}$ than exhaustive searching and check the pairs of $\mathbf{c}'$ and $\mathbf{c}''$ satisfied that the sum of $\mathbf{M}\mathbf{c}'$ and $\mathbf{M}\mathbf{c}''$ in particular entries are around $0, \pm\frac{\sqrt{2}}{\sqrt{\alpha}}$. By utilizing particular sort algorithms, the pairs that need to be checked can be greatly reduced so as the complexity.

The obstacle of TMTO strategy is that it needs substantial storage. For particular settings $n = 512, \alpha = 16$, the required storage for storing codewords of C_8 is at level EB. Considering the precision of the float number and noise in each sketching, to efficiently decrease the number of pairs to compare, the entries need to sum and sort would be more, making the storage requirement unacceptable.

Due to the significant storage demands of the TMTO strategy, we chose not to implement it, focusing instead on providing a complexity analysis. As depicted in Figure 2, when compared to the TMTO approach, our attack based on the SVD algorithm requires comparable computational resources when $\alpha \geq 16(n = 512)$ given two sketches, and less computational resources when $\alpha \geq 16(n = 512)$ given $n - 1$ sketches if $\alpha < 64$. In specific scenarios ($n = 512, \alpha = 16$), the computational requirements under no noise of our algorithm are comparable to those of the TMTO strategy, with our algorithm requiring approximately $c * 2^{60}$ multiplications versus $2^{60.8}$ additions of TMTO(c is a small constant relative to the SVD algorithm used). Moreover, our algorithm requires only a small amount of constant storage space, in contrast to the TMTO strategy, which demands large amounts of storage that are unacceptable in the proposed settings of IronMask. And the experiments in Section 5 demonstrate the effectiveness of our attacks while TMTO strategy might be not effective in same noise levels. Therefore, we contend that our algorithm is the first practical attack on IronMask in the real world.

5 Experiments

5.1 Setup

We conduct experiments to attack **IronMask** protecting **ArcFace** with specifically parameter settings($n = 512, \alpha = 16$). All experiments are carried out on a single laptop with Intel Core i7-12700H running at 2.30 GHz and 64 GB RAM. For SVD-based linear regression solver, we use the function **null_space** of python library scipy[1] and **svd** of numpy[2] library. For LSA-based linear regression solver, we implement using python and numpy library. The source code is available at [50].

5.1.1 Performance Evaluation Model. Our attack's computational complexity is characterized by three key parameters:

- r_k: Expected number of matrices sampled by linear equation sampler until it samples the "correct" matrix $\mathbf{M}$ in Definition 4.1;
- t_k: running time of linear regression solver that produces the solution of $\text{argmin}_{\mathbf{w}} ||\mathbf{M}\mathbf{w}||$ or terminates with a bot response;
- p_k: Probability that the solution obtained from linear regression solver passes the threshold determinant when the sampled matrix is "correct".

Then the expected running time t_{all} of whole algorithm can be derived as:

$$t_{all} = \frac{r_k \times t_k}{p_k}. \tag{8}$$

As r_k could be calculated by formula $r_k = \frac{1}{p_s}$(See in Theorem 4.2) given n, k, α, our task is to estimate t_k and p_k for varying number of equations k under specific scenario.

5.1.2 LSA solver optimization. The LSA-based solver employs a two-level iteration scheme:

- Outer Loop: Maximum iterations ni_{out} (terminates early if solution found);
- Inner Loop: Greedy computation with average time t_{in}.

To achieve the best performance, we constraint the outer loop of the LSA solver to a single iteration. As we have $t_k = ni_{out} * t_{in}$. Assuming that the probability that LSA-based solver produces correct template in each outer iteration is p_{out}. The probability that LSA-based solver produces correct template given "correct" matrix before ni_{out} iterations is $1 - (1 - p_{out})^{ni_{out}}$. To minimize the overall running time is equal to minimizing $\frac{ni_{out}}{1-(1-p_{out})^{ni_{out}}}$. Hence, we arrive at $ni_{out} = argmin_{ni}\frac{ni}{1-(1-p_{out})^{ni}}$. Since $0 < 1 - p < 1$, we could deduce that $ni_{out} = 1$ and thus $t_k = t_{in}$.

[1]https://scipy.org/
[2]https://numpy.org/

5.2 Experiments in Noiseless Scenario

Since r_k and t_k are both influenced by the number of sampled linear equations(k), we conduct experiments to determine the optimal setting with the shortest expected time for various values of k.

We conduct experiments on the noiseless scenario, specifically with $\theta = 0$ in Definition 3.6. The estimated running time are shown in Table 1. Our experiments indicate that

- For SVD-based probabilistic linear regression solver, the expected running time is approximately 1.7 year given only 2 sketches and 5.3 day given $n - 1 = 511$ sketches;
- For LSA-based probabilistic linear regression solver, the minimum expected running time is 4.8 day with 3 sketches and 7.1 day with 281 sketches.

The results demonstrate that our algorithms are practical to attack **IronMask** applied to protect **ArcFace** in noiseless scenario.

5.3 Experiments in Noisy Scenarios

In real-world, it's better suited that the templates sketched each time have noise between each other, i.e. $Angle(\mathbf{M}_i^{-1}\mathbf{c}_i, \mathbf{M}_j^{-1}\mathbf{c}_j) < \theta'$ where $\mathbf{c}_i, \mathbf{c}_j$ are corresponding codewords to $\mathbf{M}_i, \mathbf{M}_j$. For example, in FEI dataset[48], the angle distance between different poses(p03, p04, p05, p06, p07, p08, p11, p12) of same face is below 36° with 92% probability and below 11° with 0.3% probability.

We argue that our algorithms possess the capability to accommodate medium noise θ' with requirement of more iterations. Consequently, if the adversary have ability to choose sketches that maintain small angle distances within each corresponding original templates, they can still employ out attack to recover the original template.

However, upon consideration of noise, we discovered that even if the matrix is "correct", the threshold determinant alone cannot effectively discard solutions that deviate from the original template. This limitation arises because some candidate solutions $\mathbf{w}_{r_1}$ which are close to original template exhibit the characteristic that $\mathbf{M}_2\mathbf{w}_{r_1}$ are also close to the closest codeword, leading the algorithm to produce slight more candidate solutions. Nonetheless, there remains a high probability, denoted as p_f, that the output template of the algorithm is parallel to original template given a sampled "correct" matrix. Therefore, the expected running time t_{all} that the algorithm finally output template $\mathbf{w}$ or $-\mathbf{w}$ is revised as:

$$t_{all} = \frac{r_k \times t_k}{p_k \times p_f}. \tag{9}$$

The corresponding results are shown in Table 3.

As k increases, $r_k \times t_k$ grows exponentially, while $p_k \times p_f$ decreases exponentially. Consequently, there exists a minimum t_{all} in the mid-range of k, like the noiseless scenario in Figure 3 with $p_f = 1$. Therefore, the Table 3 represents only the approximate minimum t_{all} in noisy environments.

Table 3 reveals that our algorithms require a greater number of sampled linear equations, resulting in increased expected running time. This is because the regression problem solver needs more linear equations to obtain a correct solution due to the higher noise levels in each equation. Nevertheless, it is important to note that these algorithms are fully parallelizable. Therefore, by deploying additional machines or leveraging high-performance computing resources, we can effectively parallelize the algorithms, enabling our algorithm to recover the template within an acceptable running time, even when faced with noise levels of 36°.

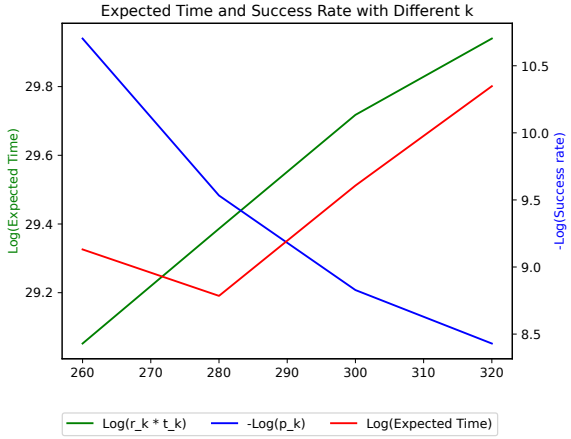

Figure 3: $Log_2(r_k * t_k), Log_2(\frac{1}{ps}), Log_2(t_{all})$ of different k using LSA-based attack algorithm when getting $k + 1$ sketches in noiseless environments. The local minimum of t_{all} is reached as $k \approx 280$.

5.4 Experiments in Real-World Dataset

Based on the previous section's findings, we have determined that the LSA-based linear regression solver exhibits superior performance. Therefore, we choose to employ the LSA-based attack algorithm, which necessitates the use of 3 sketches, for real-world simulations.

For our experiments, we choose FEI dataset and Color-FERET dataset as real-world datasets. The backbone neural network of **ArcFace** used is ResNet100 pre-trained by Insightface[3]. The FEI face database is a Brazilian face database that contains 14 images for each of 200 individuals, thus a total of 2800 images. The TPR(True Positive Rate) of selected 11 poses (p02, p03, p04, p05, p06, p07, p08, p09, p11, p12, p13) is 99.48%. We choose 2 sets of poses(p03, p05, p08 and p04, p05, p06) to simulate noisy environment and constrained environment. The Color-FERET dataset is more noisy. We choose 3 poses(fa, hl, hr) of 1123 identities to simulate the noisy environment. The TPR of Color-FERET is 89.7%.

Table 2 shows that LSA-based probabilistic attack is applicable both in noisy environment and constrained environment of FEI dataset. Even for Color-FERET dataset, the expected running time is acceptable with more machines. The results demonstrating the effectiveness of our attacks in real world.

6 Discussion of Plausible Defenses

In the following content of this section, we first analyze the obstacles to repairing the secure sketch's reusability in an information-theoretic perspective. Then we discuss two strategies aimed at mitigating the impact of potential attacks against reusability. It is

[3]https://github.com/deepinsight/insightface/tree/master/recognition/arcface_torch

Table 1: The estimated expected running time for successfully retrieving template $\mathbf{w}$ or $-\mathbf{w}$ with an expected value of 1 are analysed across different parameters k with fixed $n = 512$ for algorithms based on LSA-based solver in Algorithm 4 and SVD-based solver in Algorithm 3. Note that if the sketches are 2 for SVD-based solver, the algorithm is optimized mentioned in Section 4.2.2. t_k is approximated by computing the mean of the running time over 1000 iterations. p_k is estimated by calculating the proportion of successful times observed over maximum 20000 trials under the constraint that sampled matrix is "correct" in Definition 4.1.

Algorithm	# sketches	k	r_k	Time(t_k)	p_k	θ_t	Time(t_{all})
SVD	2	511	$2^{32.6}$	8.4 ms	100%	10°	1.7 year
	511		$2^{23.4}$	41ms	100%		5.3 day
LSA	3	220	$2^{11.35}$	102.0ms	$\frac{1}{1538.5}$	10°	4.8 day
		240	$2^{12.54}$	108.6ms	$\frac{1}{833.3}$		6.2 day
		260	$2^{13.75}$	116.0ms	$\frac{1}{512.8}$		9.5 day
		280	2^{15}	130.0ms	$\frac{1}{219.8}$		10.9 day
	261	260	$2^{11.91}$	105ms	$\frac{1}{1666.7}$	10°	7.8 day
	281	280	$2^{12.83}$	114ms	$\frac{1}{740.7}$		7.1 day
	301	300	$2^{13.74}$	123ms	$\frac{1}{454.5}$		8.86 day
	321	320	$2^{14.65}$	105ms	$\frac{1}{344.8}$		10.8 day

Table 2: Estimated expected running time using LSA-based probabilistic linear regression attacker on Real-World Dataset. $p_k * p_f$ is estimated by calculating the proportion of successful times observed over maximum 20000 trials under the constraint that sampled matrix is "correct" in Definition 4.1.

Dataset	Noise(θ')	k	r_k	Time(t_k)	$p_k * p_f$	θ_t	Time(t_{all})
FEI(p04, p05, p06)	22.56°	300	$2^{16.29}$	131.8ms	$\frac{1}{847}$	40°	103.7 day
FEI(p03, p05, p08)	28.56°	340	$2^{18.97}$	129.0ms	$\frac{1}{697}$	40°	1.47 year
Color-FERET(fa, hl, hr)	39.69°	400	$2^{23.3}$	164.6ms	$\frac{1}{654}$	49°	35.6 year

worth noting that these strategies are mutually independent, allowing us to combine them together to strengthen the HyperSphere-SS against our attacks.

6.1 Limit the Space of Secure Sketch

In Definition 3.5, the orthogonal matrix $\mathbf{M}$ does not have other constraints so that there only few pairs $(\mathbf{c}_1, \mathbf{c}_2)$ satisfies $\mathbf{c}_2 = \mathbf{M}_2\mathbf{M}_1^{-1}\mathbf{c}_1$. We may want that $\forall \mathbf{w}, \forall \mathbf{M}_1, \mathbf{M}_2 \in \mathrm{SS}(\mathbf{w}), \forall \mathbf{c} \in C, \mathbf{M}_2\mathbf{M}_1^{-1}\mathbf{c} \in C$ so that the utilization of multiple sketches does not result in any additional information leakage beyond that of a single sketch. Then our attacks will not work. It requires that $\mathbf{T} = \mathbf{M}_2\mathbf{M}_1^{-1}$ is not only an orthogonal matrix, but also maps $C \to C$. Here we give the format of matrices that maps $C_\alpha \to C_\alpha$(Proof seen in Appendix B).

Theorem 6.1. *The form of orthogonal matrices $\mathcal{T} = \{\mathbf{T} : C_\alpha \to C_\alpha\}$ in $\mathbb{R}^n$ with constraints $\alpha \neq 2$ and $\alpha \neq n$ is:*

$$\begin{pmatrix} \pm\mathbf{e}_{i_1} & \pm\mathbf{e}_{i_2} & \cdots & \pm\mathbf{e}_{i_n} \end{pmatrix} \tag{10}$$

where $\mathbf{e}_{i_1}, \mathbf{e}_{i_2}, \cdots, \mathbf{e}_{i_n}$ is a permutation of unit vectors $\mathbf{e}_0, \mathbf{e}_1, \cdots, \mathbf{e}_n$.

The remaining problem is how to choose $\mathbf{M}_1$ such that $\mathbf{TM}_1$ does not reveal too much information of template $\mathbf{w}$. One strategy is to use naive isometry rotation[26] to define $\mathbf{M}_1$ by fixing the mapped codeword. However, we find an attack that can retrieve the template with almost 60% accuracy only given $\mathbf{TM}_1$(see in Appendix B). The other strategy is to define $\mathbf{M}_1$ with randomness of $\mathbf{w}$ and hash function H, i.e. $\mathbf{M}_1 = H(\mathbf{w})$. But the problem is that if H is sensitive to the difference of $\mathbf{w}$, it's still vulnerable to our probabilistic linear regression attack in noisy scenario(see in Section 5.3). Whether suitable orthogonal matrix given $\mathbf{w}$ without leaking too much information exists remains an open problem.

6.2 Strategy 1: Add Extra Noise

Our attacks and the TMTO strategy are effective primarily because the inherent noise between each template is small, allowing us to identify a unit vector $\mathbf{w}'$ that satisfies the criterion of $Angle(\mathbf{w}', \mathbf{Rec}(\mathbf{w}', \mathbf{M}_i \in \{\mathrm{SS}(\mathbf{w})\}))$ *below a predefined threshold slightly larger than the estimated noise.* We classify these algorithms as *distance-based* algorithms. However, if the noise between each enrolled template becomes significant enough for random unit vectors to meet the same or slightly larger angle threshold criterion, the *distance-based* algorithms lose their effectiveness. A straightforward approach to increase the angle among templates is to introduce additional random noise.

6.2.1 Noise Amplification Mechanism. Assume the m noisy templates are $\{\mathbf{w}_i\}$. Let $\theta_i = Angle(\mathbf{w}_i, \mathbf{w}_j)$ represent the initial noise among the m templates, and let $\theta_a = Angle(\mathbf{w}_i, \mathbf{w}_{ia})$ denote the additional noise introduced during the sketching step, where $\mathbf{w}_{ia} = \mathbf{w}_i + \epsilon_a$ is the template with additional noise ϵ_a. Our method involves using $\mathbf{w}_{ia}$ as the enrolled template instead of $\mathbf{w}_i$. Based on Theorem 6.2, we have $E(\mathbf{w}_{ia}^T\mathbf{w}_{ja}) = \cos\theta_i \cos^2\theta_a$ and $E(\mathbf{w}_{ia}^T\mathbf{w}_j) = \cos\theta_i \cos\theta_a$.

Table 3: Estimated expected running time for different parameters θ_t, k. $p_k * p_f$ is estimated by calculating the proportion of successful times observed over maximum 20000 trials under the constraint that sampled matrix is "correct" in Definition 4.1.

Noise(θ')	Algorithm	# sketches	k	r_k	Time(t_k)	$p_k * p_f$	θ_t	Time(t_{all})
8.7°	SVD	2	522	$2^{34.6}$	7.0 ms	50%	40°	6 year
		531	531	$2^{24.32}$	33.9ms	78%		10.5 day
	LSA	3	240	$2^{12.54}$	110ms	$\frac{1}{800}$	30°	6.1 day
		321	320	$2^{14.65}$	112ms	$\frac{1}{350.9}$		11.8 day
14°	SVD	2	532	$2^{34.6}$	5.6 ms	25%	40°	18.9 year
		551	551	$2^{25.24}$	35.1ms	40%		40.2 day
	LSA	3	280	$2^{15.01}$	126ms	$\frac{1}{392}$	30°	18.8 day
		321	320	$2^{14.65}$	116ms	$\frac{1}{454.5}$		15.8 day
19°	SVD	2	552	$2^{36.57}$	5.3 ms	18%	40°	95 year
		591	591	$2^{27.06}$	36.6ms	26%		229.6 day
	LSA	3	280	$2^{15.01}$	129ms	$\frac{1}{833.4}$	30°	41 day
		321	320	$2^{14.65}$	115ms	$\frac{1}{1176.5}$		40.5 day
26°	SVD	2	592	$2^{40.8}$	4.4 ms	10%	40°	2676 year
		671	671	$2^{30.73}$	38.5ms	13.6%		16 year
	LSA	3	320	$2^{17.61}$	129ms	$\frac{1}{645.2}$	40°	193 day
		381	380	$2^{17.4}$	155ms	$\frac{1}{1111.1}$		346 day
30°	SVD	771	771	$2^{41.72}$	45.9 ms	42.2%	40°	147 year
	LSA	3	340	$2^{18.97}$	132ms	$\frac{1}{869.6}$	40°	1.87 year
		421	420	$2^{19.2}$	155ms	$\frac{1}{1428.6}$		4.34 year
36°	SVD	911	911	$2^{45.3}$	49.5 ms	13.4%	45°	4.26×10^4 year
	LSA	3	380	$2^{21.82}$	145ms	$\frac{1}{2857.1}$	45°	48.65 year
		521	520	$2^{23.82}$	178ms	$\frac{1}{2222.2}$		185 year
43°	LSA	3	440	$2^{21.82}$	168ms	$\frac{1}{10000}$	50°	4.8×10^3 year
		681	680	$2^{23.82}$	225ms	$\frac{1}{4851.75}$		8.2×10^4 year

For the user, the template $\mathbf{w}_j$ is utilized to retrieve the enrolled template $\mathbf{w}_{ia}$, resulting in noise approximately $\cos\theta_i \cos\theta_a$. In contrast, for the attacker, the noise between each enrolled template is $\cos\theta_i \cos^2\theta_a$. The asymmetry in noise levels allows the user to still successfully retrieve the template, while significantly hindering the attacker's ability to carry out attacks due to the presence of larger noise.

Theorem 6.2. *Assume $\{\mathbf{w}_i, i \in [4]\}$ be four unit vectors in a space of dimension $n \geq 3$. Define $\theta_{ij} = Angle(\mathbf{w}_i, \mathbf{w}_j)$, $\mathbf{v} = unit(\mathbf{w}_2 - \mathbf{w}_2^T\mathbf{w}_1 \cdot \mathbf{w}_1)$ and $\mathbf{u} = unit(\mathbf{w}_4 - \mathbf{w}_3^T\mathbf{w}_4 \cdot \mathbf{w}_3)$, where $unit(\mathbf{r}) = \frac{\mathbf{r}}{\|\mathbf{r}\|}$. If u is a random unit vector orthogonal to $\mathbf{w}_1$ and v is a random unit vector orthogonal to $\mathbf{w}_3$, then $E(\cos\theta_{24}) = \cos\theta_{12}\cos\theta_{13}\cos\theta_{34}$.*

6.2.2 Quantitative Noise Determination. The remaining problem is determining how much noise should be introduced. To address this, we propose Hypothesis 1.

Hypothesis 1. *Let $\theta_i = Angle(\mathbf{w}_i, \mathbf{w}_j)$ denote the initial noise among the m templates. Let $\theta_a = Angle(\mathbf{w}_i, \mathbf{w}_{ia})$ denote the additional noise introduced during the sketching step, where $\mathbf{w}_{ia} = \mathbf{w}_i + \epsilon_a$. Further, let $\theta_r = \cos^{-1} \mathrm{E}(\mathbf{r}^T\mathbf{Decode}(\mathbf{r}))$, where $\mathbf{Decode}$ is ECC's decoding algorithm and $\mathbf{r}$ is a random unit vector. We hypothesize that if $E(\mathbf{w}_{ia}, \mathbf{w}_{ja} | i \neq j) = \cos\theta_i \cos^2\theta_a = \cos\theta_r$, then "distance-based" algorithms that identify a unit vector $\mathbf{w}'$ satisfying the criterion of "$Angle(\mathbf{w}', \mathbf{Rec}(\mathbf{w}', \mathbf{M}_i \in \{\mathbf{SS}(\mathbf{w})\}))$ below a predefined threshold slightly larger than the estimated noise" perform no better than brute-force algorithm.*

Particularly, in the scenario where an attacker gets two sketches $\mathbf{M}_1 = \mathbf{SS}(\mathbf{w}_{1a} = \mathbf{w}_1 + \epsilon)$ and $\mathbf{M}_2 = \mathbf{SS}(\mathbf{w}_{2a} = \mathbf{w}_2 + \epsilon)$, the "distance-based" algorithm would identify numerous "unit vectors" satisfying its criterion. This gives us confidence of the validity of the hypothesis. As for a random codeword $c_1' \in C_\alpha$, we have:

$$\begin{aligned}\theta_r &= Angle(\mathbf{r}, \mathbf{Decode}(\mathbf{r})) \\ &\approx Angle(\mathbf{M}_2\mathbf{M}_1^{-1}\mathbf{c}_1', \mathbf{Decode}(\mathbf{M}_2\mathbf{M}_1^{-1}\mathbf{c}_1')) \\ &= Angle(\mathbf{M}_1^{-1}\mathbf{c}_1', \mathbf{M}_2^{-1}\mathbf{Decode}(\mathbf{M}_2\mathbf{M}_1^{-1}\mathbf{c}_1')) \\ &= Angle(\mathbf{M}_1^{-1}\mathbf{c}_1', \mathbf{Rec}(\mathbf{M}_1^{-1}\mathbf{c}_1', \mathbf{M}_2)),\end{aligned}$$

assuming that $\mathbf{M}_2\mathbf{M}_1'\mathbf{c}_1'$ can be treated as a random vector. This assumption holds because $\mathbf{M} = \mathbf{M}_2\mathbf{M}_1'$ only contains the constraint of mapping $\mathbf{c}_1$ to $\mathbf{c}_2$ with noise θ_r, while a random matrix $\mathbf{M}_r$ would also satisfy this constraint. Thus we could see $\mathbf{M}$ indistinguishable from random matrices, and $\mathbf{M}_2\mathbf{M}_1'\mathbf{c}_1'$ could be treated as a random vector. This justifies the approximation.

Next, assuming $\mathbf{c}_2' = \mathbf{M}_2\mathbf{Rec}(\mathbf{M}_1^{-1}\mathbf{c}_1', \mathbf{M}_2)$, we derive:

$$Angle(\mathbf{w}_{1a}, \mathbf{w}_{2a}) \approx Angle(\mathbf{M}_1^{-1}c_1', \mathbf{M}_2^{-1}c_2')$$

$$\mathbf{w}' = unit(\mathbf{M}_1^{-1}c_1' + \mathbf{M}_2^{-1}c_2'), \mathbf{w} = unit(\mathbf{w}_{1a}, \mathbf{w}_{2a})$$

$$Angle(\mathbf{w}', \mathbf{Rec}(\mathbf{w}', \mathbf{M}_i \in \{\mathbf{M}_1, \mathbf{M}_2\})) \approx$$
$$Angle(\mathbf{w}, \mathbf{Rec}(\mathbf{w}, \mathbf{M}_i \in \{\mathbf{M}_1, \mathbf{M}_2\}))$$

Thus, $\mathbf{w}'$ is indistinguishable from $\mathbf{w}$ under the criterion of the *distance-based* algorithm.

However, this perfect property does not extend to scenarios involving more than two sketches. Because the distance between the noisy templates and their geometric center is smaller than the distance between any set $\{\mathbf{M}_i^{-1}\mathbf{c}'_i\}$ and their geometric center. Nevertheless, based on our empirical observations(particularly Table 3), *distance-based* algorithms do not significantly benefit from much more additional sketches and may even perform worse in some cases. Given that the hypothesis is strongly supported in the two-sketch scenario, we are confident that it holds for other *distance-based* algorithms and more-sketch scenario as well.

*6.2.3 Application in **IronMask**.* Under the prerequisite of Hypothesis 1, the relationship between the initial noise level and the success rate of the secure sketch recovery algorithm is in Table 4. The table shows that as α decreases, while maintaining the same recovery success probability p_r, the initial noise θ_i increases. Therefore, by fixing p_r, we can illustrate the relationship between θ_i and α in Figure 4.

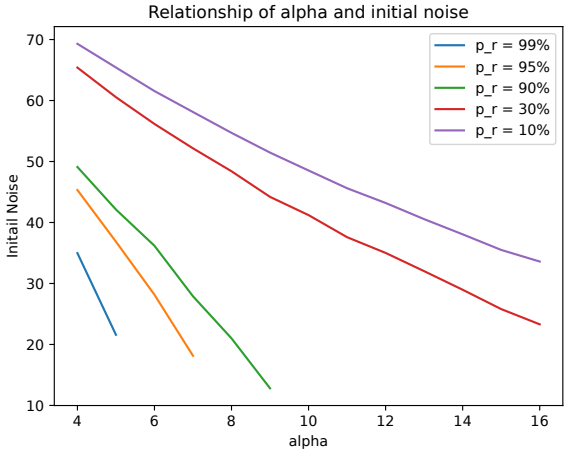

Figure 4: The relationship of initial noise θ_i and secure sketch parameter $\alpha(n = 512)$ with different recovery probability p_r.

Assume an initial noise level of 36°, which is the criterion for the FEI dataset, to maintain a recovery probability of approximately 90%, we find that $\alpha \leq 6$. This implies the brute-force space is reduced to less than $2^{50.46}$. However, if the attacker can obtain sketches from closer templates, the initial noise level should be lower than the criterion. For instance, in the FEI dataset, templates from poses p04, p05 and p06 are closer to each other than other poses. In such cases, the initial noise is 22.56°. To maintain the recovery probability high, we set $\alpha = 5$, achieving a recovery probability $p_r = 95\%$ in the FEI dataset with poses(p03, p04, p05, p06, p07, p08, p11, p12). Here, the size of the brute-force attack space is reduced from 2^{115} to 2^{43}.

6.3 Strategy 2:Salt

To carry out attacks targeting reusability, a minimum of two protected templates is necessary. By incorporating randomness into our protection algorithm, we can effectively slow down the attack algorithm's progress. In the context of the HyperSphere-SS algorithm, we propose appending n_f additional random matrices,

Table 4: The relation between initial noise(θ_i) and success rate(p_r) of recovery algorithm in secure sketch under different α with $n = 512$. θ_r is the prerequisite of Hypthosis 1.

α	θ_i	θ_a	p_r	θ_r	$\log_2(\lvert C_\alpha \rvert)$
16	0°	48.3°	56%	63.8°	115
	23.3°	46.1°	30%		
	33.6°	43.2°	10%		
12	0°	50.9°	76%	66.6°	91
	34.8°	45.9°	30%		
	43.2°	42.3°	10%		
8	0°	54.3°	95%	70.1°	64
	21.0°	52.8°	90%		
	48.4°	44.2°	30%		
	54.7°	39.8°	10%		
4	0°	59.4°	100%	75.0°	35
	49.1°	51.1°	90%		
	65.4°	38.2°	30%		
	69.3°	31.2°	10%		

independent of the template $\mathbf{w}$, along with the output sketch matrix $\mathbf{M}$. Subsequently, the recovery process is modified to invoke recovery algorithm with these $n_f + 1$ matrices and take the recovery template closest to the input template as the final output. The revised secure sketch's algorithms are as below:

- Sketching algorithm $\mathbf{SS}_{\mathbf{salt}}$: on input $\mathbf{w} \in S^{n-1}$, $\mathbf{c} \xleftarrow{\$} C$, randomly generate an orthogonal matrix $\mathbf{M}$ that satisfies $\mathbf{c} = \mathbf{Mw}$. Additionally, randomly generate n_f orthogonal matrices $\{\mathbf{M}_i^r\}$. Output set $\{\mathbf{M}, \mathbf{M}_i^r\}$ as sketch;
- Recovery algorithm $\mathbf{Rec}_{\mathbf{salt}}$: on input $\mathbf{w}' \in S^{n-1}$, orthogonal matrices set $S = \{\mathbf{M}\}$, calculate set $W_c = \{\mathbf{Rec}(\mathbf{w}', \mathbf{M}\}$ where $\mathbf{Rec}$ is the original recovery algorithm of the secure sketch. Output $\mathbf{w}'' = \mathrm{argmin}_{\mathbf{w}'' \in W_c} Angle(\mathbf{w}'', \mathbf{w}')$.

Without ambiguity regarding the original sketch, we refer to the output of $\mathbf{SS}_{\mathbf{salt}}$ as a sketch-set. Intuitively, given m sketch-sets $\{S_i \in \mathbf{SS}_{\mathbf{salt}}(\mathbf{w}), i \in [m]\}$, if an attacker cannot efficiently identify the correct sketches from each sketch-set S_i, it would require $O((n_f + 1)^m)$ attempts to sample the correct sketches and complete the attack. This leads to the Hypothesis 2.

Hypothesis 2. *Assume $\mathbf{w}$ as original template, σ_i as the ith noise, $\mathbf{M}^i = \mathbf{SS}(\mathbf{w} + \sigma_i)$, $S_f^i = \{\mathbf{M}_j^i | j = 1, \ldots, n_f\}$ as corresponding additional random matrices , $S^i = \{\mathbf{M}^i\} \cup S_f^i$ and the attacker's algorithm for targeting reusability of original secure sketch algorithm is $\mathcal{A}$. Then we have algorithm $\mathcal{A}'$ as followings:*

(1) *Set $S = \{\mathbf{M}^{i'} | \mathbf{M}^{i'} \xleftarrow{\$} S^i\}$;*
(2) *$\mathbf{w} \leftarrow \mathcal{A}(S)$;*
(3) *Calculate $d = \sum_{\mathbf{M} \in S} ||\mathbf{Rec}(\mathbf{w}, \mathbf{M}) - \mathbf{w}||$, if $d >$ threshold, output S; otherwise, return to step 1.*

Assume the algorithms that takes $\{S^i\}$ as input and outputs a set of sketches as S_a. We hypothesize that there are no more efficient algorithms than $\mathcal{A}'$ that satisfies $\mathcal{A}'(\{S^i\}) = \{M^i\}$.

Assume the Hypothesis 2 holds, from the attacker's perspective, acquiring m protected templates, each containing $n_f + 1$ matrices,

necessitates the examination of $(n_f + 1)^m$ matrix pairs to employ the original attack algorithm. Consequently, we enhance the attack algorithm's complexity to at least a factor of $O(n_f^2)$ at the cost of $O(n_f)$ times additional computations in the recovery algorithm. In computational view, this enhancement is equal to augmenting security by $\log_2(n_f)$ security bits.

6.4 Combination of Adding Extra Noise and Salting

To integrate salting method with the adding extra noise approach, a trade-off between recovery time and security strength must be carefully considered.

For FEI dataset, if we take $\alpha = 5$ in Section 6.2, the average runtime for recovery algorithm in a laptop with an i7-12700h Intel Core processor is $22.37\mu s$. To maintain the recovery time acceptable, we could take $n_f = 2^{16.4}$. Then the time for recovery is approximate $1s$. while for the brute-force attacker that exhaustively searches the codeword space to find the codeword pairs $(\mathbf{c}_i, \mathbf{c}_j)$ satisfying $Angle(\mathbf{M}_j^T \mathbf{M}_i \mathbf{c}_i, \mathbf{c}_j) < threshold$, the average runtime for invoking recovery algorithm in a single attempt is $22.37\mu s/512 * 5 \approx 0.218\mu s$. Then the whole time is estimated as $22.37\mu s/512 * 5 * 2^{43} * 2^{16.4*2}/2 \approx 2.28 * 10^8$ year, making the attack infeasible.

Due to the codeword space is shrinking to 2^{43}, the fuzzy commitment scheme is not secure enough for directly solving $\mathbf{c}$ from $H(\mathbf{c})$ where $\mathbf{c} \in C_\alpha$. To enlarge the search space of hash function, we revise the commitment to be $H(\mathbf{c}, \mathbf{M})$ to extend the search space from C_α to $C_\alpha \times \mathcal{S}_\mathbf{M}$ where $\mathbf{M}$ is the sketch of $\mathbf{c}$ and $\mathcal{S}_\mathbf{M}$ contains $\mathbf{M}$ and n_f additional matrices. Thus for FEI dataset, the size of brute-force attacker's search space is $2^{16.4+43} = 2^{59.4}$. And if we set the time of hash function approximate 2s, the complexity for directly attacking hash function is comparable to attacking secure sketch.

In summary, by combining these two approaches, we have enhanced the HyperSphere-SS scheme against reusability attacks and achieved brute-force security of $2^{59.4}$. However, this comes at the cost of slower authentication time (approximately $1 + 2 = 3$s) and significant storage requirements (specifically, $2^{16.4} \times 32 \times 512^2 \approx 84.4GB$). Alternative methods to alleviate storage burden, such as generating the sektch-set on-the-fly, are not practical due to the excessive time required to generate $84.4GB$ of pseudorandom bitstreams (e.g., OpenSSL generates random bitstreams at a speed of 1GB/s). Therefore, we do not adopt this approach.

7 Influence to Other BTP Scheme

Our work can be regarded as a class of methods for solving linear properties derived from protected biometric templates. In the works [4, 29], it was discovered that if at least n linearly independent relations with noise can be derived from the protected templates (where n is the dimension of the embedded space), then the template protection algorithm does not satisfy reusability. Specifically, for a template protection algorithm S, if for a template $\mathbf{w}$ and a set P of protected templates from $S(\mathbf{w})$, there exists:

$$L = \{\mathbf{v}_i | \mathbf{v}_i^T \mathbf{w} \approx \epsilon_i, \{\mathbf{v}_i \text{ are independent}\} \leftarrow \mathbf{X}(P)\}, |L| \geq n,$$

where $\mathbf{X}$ is an determinant efficient algorithm and ϵ_i is small noise. Then the algorithm S does not satisfy reusability.

Our work extends the attack surface, meaning that for a template protection algorithm S, if there exists:

$$L' = \{\mathbf{v}_i | \mathbf{v}_i^T \mathbf{w} \approx \epsilon_i, \{\mathbf{v}_i \text{ are independent}\} \leftarrow \mathbf{X}(P)\}, |L'| \geq n^*,$$

where $\mathbf{X}$ is a probabilistic algorithm(may not so efficient) and ϵ_i is small noise. Then the algorithm S does not satisfy reusability. Note that we may only require $|L'| \geq n^*$ instead of $|L'| \geq n$ because we can leverage nonlinear structural properties of the template (e.g., the sparsity of the codewords in our case) to reduce the number of linear relations that need to be sampled.

Boyen [4] proves that template protection algorithms based on nonlinear error-correcting codes do not satisfy reusability; the algorithm proposed by Kim et al. [29] can be used to attack the reusability security of LSA-based template protection algorithms, such as [21]. Our work can be used to attack the reusability security of template protection methods based on the hypersphere error-correcting code construction in IronMask, including [26, 28]. This inspires us to consider whether protected templates may probabilistically leak linear relations. If so, the reusability security of the template protection scheme may be compromised.

We emphasize that fuzzy extractors based on linear error correcting codes, such as those in [4, 49], and schemes based on the Sample-Then-Lock approach [7] are not affected by our attacks. This is because, for the former, $|L|$ is a fixed value less than n for any algorithm $\mathbf{P}$; for the latter, the values of linear relations are hidden using digital lockers, making it computationally infeasible to derive linear relations.

8 Conclusion

IronMask conceptualized as fuzzy commitment scheme is to protect the face template extracted by ArcFace in hypersphere, claims that it can provide at least 115-bit security against previous known attacks with great recognition performance. Targeting on renewability and unlinkability of **IronMask**, we proposed probabilistic linear regression attack that can successfully recover the original face template by exploiting the linearity of underlying used error correcting code. Under the recommended parameter settings on **IronMask** applied to protect **ArcFace**, our attacks are applicable in practical time verified by our experiments, even with the consistent noise level across biometric template extractions. To mitigate the impact of our attacks, we propose two plausible strategies for enhancing the hypersphere secure sketch scheme in **IronMask**. However, these strategies come at the cost of reduced security level(approximately 60 bits), lower recovery success probabilities (95%), slower authentication (3 seconds) and significant storage requirement (~80GB)., which may hinder their practical usability. To fully alleviate the error correcting code capability, future designs of hypersphere secure sketches and error-correcting codes should carefully consider the potential linearity of codewords, which may render them susceptible to attacks like the one we've presented.

Acknowledgments

This study was funded by the National Natural Science Foundation of China (62372294). And thanks for GPT-like tools, including DeepSeek R1, DeepSeek V3 and Ernie Bot, to polish the writings of whole paper with prompts such as "please polish these sentences",

"please polish these academic paragraphs, maintain the sentences' structure and focusing on grammatical wording", "polish the sentence and make is still simple" and etc.

References

[1] Meng Ao and Stan Z. Li. 2009. Near Infrared Face Based Biometric Key Binding. In *Advances in Biometrics (Lecture Notes in Computer Science)*, Massimo Tistarelli and Mark S. Nixon (Eds.). Springer, Berlin, Heidelberg, 376–385. https://doi.org/10.1007/978-3-642-01793-3_39

[2] Daniel Apon, Chongwon Cho, Karim Eldefrawy, and Jonathan Katz. 2017. Efficient, Reusable Fuzzy Extractors from LWE. In *Cyber Security Cryptography and Machine Learning (Lecture Notes in Computer Science)*, Shlomi Dolev and Sachin Lodha (Eds.). Springer International Publishing, Cham, 1–18. https://doi.org/10.1007/978-3-319-60080-2_1

[3] Arathi Arakala, Jason Jeffers, and K. J. Horadam. 2007. Fuzzy Extractors for Minutiae-Based Fingerprint Authentication. In *Advances in Biometrics (Lecture Notes in Computer Science)*, Seong-Whan Lee and Stan Z. Li (Eds.). Springer, Berlin, Heidelberg, 760–769. https://doi.org/10.1007/978-3-540-74549-5_80

[4] Xavier Boyen. 2004. Reusable Cryptographic Fuzzy Extractors. In *Proceedings of the 11th ACM Conference on Computer and Communications Security*. ACM, Washington DC USA, 82–91. https://doi.org/10.1145/1030083.1030096

[5] T. Tony Cai and Lie Wang. 2011. Orthogonal Matching Pursuit for Sparse Signal Recovery With Noise. *IEEE Transactions on Information Theory* 57, 7 (July 2011), 4680–4688. https://doi.org/10.1109/TIT.2011.2146090

[6] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. 2006. Stable Signal Recovery from Incomplete and Inaccurate Measurements. *Communications on Pure and Applied Mathematics* 59, 8 (2006), 1207–1223. https://doi.org/10.1002/cpa.20124

[7] Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam Smith. 2021. Reusable fuzzy extractors for low-entropy distributions. *Journal of Cryptology* 34 (2021), 1–33.

[8] R. Cappelli, D. Maio, A. Lumini, and D. Maltoni. 2007. Fingerprint Image Reconstruction from Standard Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 9 (Sept. 2007), 1489–1503. https://doi.org/10.1109/TPAMI.2007.1087

[9] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. 1998. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing* 20, 1 (1998), 33–61. https://doi.org/10.1137/S1064827596304010 arXiv:https://doi.org/10.1137/S1064827596304010

[10] S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. IEEE, San Diego, CA, USA, 539–546. https://doi.org/10.1109/CVPR.2005.202

[11] Matthijs J. Coster, Antoine Joux, Brian A. LaMacchia, Andrew M. Odlyzko, Claus-Peter Schnorr, and Jacques Stern. 1992. Improved Low-Density Subset Sum Algorithms. *computational complexity* 2, 2 (June 1992), 111–128. https://doi.org/10.1007/BF01201999

[12] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. 2022. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 10 (Oct. 2022), 5962–5979. https://doi.org/10.1109/TPAMI.2021.3087709 arXiv:1801.07698 [cs]

[13] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. 2008. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM J. Comput.* 38, 1 (Jan. 2008), 97–139. https://doi.org/10.1137/060651380 arXiv:cs/0602007

[14] Simon Foucart and Holger Rauhut. 2013. *A Mathematical Introduction to Compressive Sensing*. Springer, New York, NY. https://doi.org/10.1007/978-0-8176-4948-7

[15] Javier Galbally, Arun Ross, Marta Gomez-Barrero, Julian Fierrez, and Javier Ortega-Garcia. 2013. Iris Image Reconstruction from Binary Templates: An Efficient Probabilistic Approach Based on Genetic Algorithms. *Computer Vision and Image Understanding* 117, 10 (Oct. 2013), 1512–1525. https://doi.org/10.1016/j.cviu.2013.06.003

[16] Steven D. Galbraith and Lukas Zobernig. 2019. Obfuscated Fuzzy Hamming Distance and Conjunctions from Subset Product Problems. In *Theory of Cryptography*, Dennis Hofheinz and Alon Rosen (Eds.). Vol. 11891. Springer International Publishing, Cham, 81–110. https://doi.org/10.1007/978-3-030-36030-6_4

[17] David Gamarnik, Eren C. Kızıldağ, and Ilias Zadik. 2021. Inference in High-Dimensional Linear Regression via Lattice Basis Reduction and Integer Relation Detection. *IEEE Transactions on Information Theory* 67, 12 (Dec. 2021), 8109–8139. https://doi.org/10.1109/TIT.2021.3113921 arXiv:1910.10890 [math, stat]

[18] David Gamarnik and Ilias Zadik. 2019. Sparse High-Dimensional Linear Regression. Algorithmic Barriers and a Local Search Algorithm. arXiv:1711.04952 [math, stat]

[19] Abhishek Jana, Bipin Paudel, Md. Kamruzzaman Sarker, Monireh Ebrahimi, Pascal Hitzler, and George T. Amariucai. 2022. Neural Fuzzy Extractors: A Secure Way to Use Artificial Neural Networks for Biometric User Authentication. *Proc. Priv. Enhancing Technol.* 2022, 4 (2022), 86–104. https://doi.org/10.56553/POPETS-2022-0100

[20] Mingming Jiang, Shengli Liu, You Lyu, and Yu Zhou. 2023. Face-Based Authentication Using Computational Secure Sketch. *IEEE Transactions on Mobile Computing* 22, 12 (2023), 7172–7187. https://doi.org/10.1109/TMC.2022.3207830

[21] Zhe Jin, Jung Yeon Hwang, Yen-Lung Lai, Soohyung Kim, and Andrew Beng Jin Teoh. 2018. Ranking-Based Locality Sensitive Hashing-Enabled Cancelable Biometrics: Index-of-Max Hashing. *IEEE Transactions on Information Forensics and Security* 13, 2 (2018), 393–407. https://doi.org/10.1109/TIFS.2017.2753172

[22] Ari Juels and Madhu Sudan. 2006. A Fuzzy Vault Scheme. *Designs, Codes and Cryptography* 38, 2 (Feb. 2006), 237–257. https://doi.org/10.1007/s10623-005-6343-z

[23] Ari Juels and Martin Wattenberg. 1999. A Fuzzy Commitment Scheme. In *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS '99)*. Association for Computing Machinery, New York, NY, USA, 28–36. https://doi.org/10.1145/319709.319714

[24] Shuichi Katsumata, Takahiro Matsuda, Wataru Nakamura, Kazuma Ohara, and Kenta Takahashi. 2021. Revisiting Fuzzy Signatures: Towards a More Risk-Free Cryptographic Authentication System Based on Biometrics. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21)*. Association for Computing Machinery, New York, NY, USA, 2046–2065. https://doi.org/10.1145/3460120.3484586

[25] Christof Kauba, Simon Kirchgasser, Vahid Mirjalili, Andreas Uhl, and Arun Ross. 2021. Inverse Biometrics: Generating Vascular Images From Binary Templates. *IEEE Transactions on Biometrics, Behavior, and Identity Science* 3, 4 (Oct. 2021), 464–478. https://doi.org/10.1109/TBIOM.2021.3073666

[26] Sunpill Kim, Yunseong Jeong, Jinsu Kim, Jungkon Kim, Hyung Tae Lee, and Jae Hong Seo. 2021. IronMask: Modular Architecture for Protecting Deep Face Template. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Nashville, TN, USA, 16120–16129. https://doi.org/10.1109/CVPR46437.2021.01586

[27] Sunpill Kim, Yunseong Jeong, Jinsu Kim, Jungkon Kim, Hyung Tae Lee, and Jae Hong Seo. 2021. IronMask: Modular Architecture for Protecting Deep Face Template. arXiv:2104.02239 [cs]

[28] Sunpill Kim, Hoyong Shin, and Jae Hong Seo. 2025. Deep face template protection in the wild. *Pattern Recognition* 162 (2025), 111336. https://doi.org/10.1016/j.patcog.2024.111336

[29] Sunpill Kim, Yong Kiam Tan, Bora Jeong, Soumik Mondal, Khin Mi Mi Aung, and Jae Hong Seo. 2024. Scores Tell Everything about Bob: Non-adaptive Face Reconstruction on Face Recognition Systems . In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 1684–1702. https://doi.org/10.1109/SP54263.2024.00161

[30] J. C. Lagarias and A. M. Odlyzko. 1985. Solving Low-Density Subset Sum Problems. *J. ACM* 32, 1 (Jan. 1985), 229–246. https://doi.org/10.1145/2455.2461

[31] Youn Joo Lee, Kwanghyuk Bae, Sung Joo Lee, Kang Ryoung Park, and Jaihie Kim. 2007. Biometric Key Binding: Fuzzy Vault Based on Iris Images. In *Advances in Biometrics (Lecture Notes in Computer Science)*, Seong-Whan Lee and Stan Z. Li (Eds.). Springer, Berlin, Heidelberg, 800–808. https://doi.org/10.1007/978-3-540-74549-5_84

[32] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. 2017. SphereFace: Deep Hypersphere Embedding for Face Recognition. https://arxiv.org/abs/1704.08063v4.

[33] Guangcan Mai, Kai Cao, Pong C. Yuen, and Anil K. Jain. 2019. On the Reconstruction of Face Images from Deep Face Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 5 (May 2019), 1188–1202. https://doi.org/10.1109/TPAMI.2018.2827389 arXiv:1703.00832 [cs]

[34] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. 2021. MagFace: A Universal Representation for Face Recognition and Quality Assessment. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Nashville, TN, USA, 14220–14229. https://doi.org/10.1109/CVPR46437.2021.01400

[35] Deen Dayal Mohan, Nishant Sankaran, Sergey Tulyakov, Srirangaraj Setlur, and Venu Govindaraju. 2019. Significant Feature Based Representation for Template Protection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, Long Beach, CA, USA, 2389–2396. https://doi.org/10.1109/CVPRW.2019.00293

[36] Kathleen Moriarty, Burt Kaliski, and Aneas Rusch. 2017. *PKCS #5: Password-Based Cryptography Specification Version 2.1*. Request for Comments RFC 8018. Internet Engineering Task Force. https://doi.org/10.17487/RFC8018

[37] Hatef Otroshi Shahreza, Vedrana Krivokuća Hahn, and Sébastien Marcel. 2024. Vulnerability of State-of-the-Art Face Recognition Models to Template Inversion Attack. *IEEE Transactions on Information Forensics and Security* 19 (2024), 4585–4600. https://doi.org/10.1109/TIFS.2024.3381820

[38] Hatef Otroshi Shahreza and Sébastien Marcel. 2023. Face Reconstruction from Facial Templates by Learning Latent Space of a Generator Network. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., New

Orleans, LA, USA, 12703–12720. https://proceedings.neurips.cc/paper_files/paper/2023/file/29e4b51d45dc8f534260adc45b587363-Paper-Conference.pdf
[39] Rohit Kumar Pandey, Yingbo Zhou, Bhargava Urala Kota, and Venu Govindaraju. 2016. Deep Secure Encoding for Face Template Protection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, Las Vegas, NV, USA, 77–83. https://doi.org/10.1109/CVPRW.2016.17
[40] Colin Percival and Simon Josefsson. 2016. *The Scrypt Password-Based Key Derivation Function.* Request for Comments RFC 7914. Internet Engineering Task Force. https://doi.org/10.17487/RFC7914
[41] P. Jonathon Phillips, Harry Wechsler, Jeffery Huang, and Patrick J. Rauss. 1998. The FERET Database and Evaluation Procedure for Face-Recognition Algorithms. *Image and Vision Computing* 16, 5 (April 1998), 295–306. https://doi.org/10.1016/S0262-8856(97)00070-X
[42] Niels Provos and David Mazières. 1999. A Future-Adaptable Password Scheme. In *1999 USENIX Annual Technical Conference (USENIX ATC 99)*. USENIX Association, Monterey, CA, 81–91. http://www.usenix.org/events/usenix99/provos.html
[43] Christian Rathgeb, Johannes Merkle, Johanna Scholz, Benjamin Tams, and Vanessa Nesterowicz. 2022. Deep Face Fuzzy Vault: Implementation and Performance. *Computers & Security* 113 (Feb. 2022), 102539. https://doi.org/10.1016/j.cose.2021.102539
[44] C. P. Schnorr and M. Euchner. 1994. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math. Program.* 66, 2 (sep 1994), 181–199. https://doi.org/10.1007/BF01581144
[45] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, Boston, MA, USA, 815–823. https://doi.org/10.1109/CVPR.2015.7298682
[46] Koen Simoens, Pim Tuyls, and Bart Preneel. 2009. Privacy Weaknesses in Biometric Sketches. In *2009 30th IEEE Symposium on Security and Privacy*. IEEE, Oakland, CA, USA, 188–203. https://doi.org/10.1109/SP.2009.24
[47] Veeru Talreja, Matthew C. Valenti, and Nasser M. Nasrabadi. 2019. Zero-Shot Deep Hashing and Neural Network Based Error Correction for Face Template Protection. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, Tampa, FL, USA, 1–10. https://doi.org/10.1109/BTAS46853.2019.9185979
[48] Carlos Eduardo Thomaz and Gilson Antonio Giraldi. 2010. A New Ranking Method for Principal Components Analysis and Its Application to Face Image Analysis. *Image and Vision Computing* 28, 6 (June 2010), 902–913. https://doi.org/10.1016/j.imavis.2009.11.005
[49] Yunhua Wen, Shengli Liu, and Shuai Han. 2018. Reusable Fuzzy Extractor from the Decisional Diffie–Hellman Assumption. *Designs, Codes and Cryptography* 86, 11 (Nov. 2018), 2495–2512. https://doi.org/10.1007/s10623-018-0459-4
[50] Pengxu Zhu and Lei Wang. 2025. probabilistic-linear-regression-attack-hypersphere-secure-sketch. GitHub repository. https://github.com/page0egap/probabilistic-linear-regression-attack-hypersphere-secure-sketch

A TMTO Strategy

In [27], they describe a time-memory trade-off(TMTO) strategy to attack IronMask, here we revise the details of TMTO strategy to make it more applicable in real world's settings.

Assume $\mathbf{M}_1$ and $\mathbf{M}_2$ are two sketches of biometric template $\mathbf{w}$, our target is to find codeword pair $(\mathbf{c}_1, \mathbf{c}_2)$ in C_α such that $\mathbf{c}_2 = \mathbf{M}\mathbf{c}_1$ for orthogonal matrix $\mathbf{M} = \mathbf{M}_2\mathbf{M}_1^{-1}$. Let $\mathbf{c_1} = (c_{11}, c_{12}, \ldots, c_{1n})$, $\mathbf{c_2} = (c_{21}, c_{22}, \ldots, c_{2n})$ and $\mathbf{M} = (m_{ij})$. The equation can be rewritten as

$$c_{11}m_{i1} + c_{12}m_{i2} + \cdots c_{1n}m_{in} = c_{2i}, \forall 1 \le i \le n \tag{11}$$

. As each codeword $\mathbf{c} \in C_\alpha$ can be written as two components $\mathbf{a}, \mathbf{b}$ where each has exactly $\frac{\alpha}{2}$ non-zero elements and there is no positions that both $\mathbf{a}$ and $\mathbf{b}$ are non-zero. We can rewrite Equation(11) as

$$\sum_{\mathbf{a}=(a_1,\ldots,a_n)\in C_{\frac{\alpha}{2}}} a_j m_{ij} + \sum_{\mathbf{b}=(b_1,\ldots,b_n)\in C_{\frac{\alpha}{2}}} b_j m_{ij} = \sqrt{2}c_{2i}, \forall 1 \le i \le n \tag{12}$$

with constraint $\sqrt{2}\mathbf{c}_1 = \mathbf{a} + \mathbf{b}$. If we relax the constraint to that $\mathbf{a}$ and $\mathbf{b}$ have exactly $\frac{\alpha}{2}$ non-zero elements, the Equation(12) can be simplified as

$$\sum_{\mathbf{a}\in C_{\frac{\alpha}{2}}} a_j m_{ij} = \sqrt{2}c_{2i} - \sum_{\mathbf{b}\in C_{\frac{\alpha}{2}}} b_j m_{ij}, \forall 1 \le i \le n \tag{13}$$

. As there are three elements $-\frac{1}{\sqrt{\alpha}}, 0, \frac{1}{\sqrt{\alpha}}$ for c_{2i} and with high probability $c_{2i} = 0$ if $\alpha \ll n$, we can just assume $c_{2i} = 0$ for random selected i. Then we calculate all $t^i_{\mathbf{a}} = \sum_{\mathbf{a}\in C_{\frac{\alpha}{2}}} a_j m_{ij}$ for some i, search them and find pairs that satisfies $t^i_{\mathbf{a}} + t^i_{\mathbf{b}} = 0$ for $\mathbf{a}, \mathbf{b} \in C_{\frac{\alpha}{2}}$. As $t^i_{\mathbf{a}} = -t^i_{-\mathbf{a}}$, it only needs to find $\mathbf{a}, \mathbf{b} \in C_{\frac{\alpha}{2}}$ such that $t^i_{\mathbf{a}} = t^i_{\mathbf{b}}$. The possible codeword $\mathbf{c}_1$ is equal to $\frac{1}{\sqrt{2}}(\mathbf{a} - \mathbf{b})$.

In real world, since $t_{\mathbf{a}}$ is float number with limited precision and there's some noise in c_{2i} in real settings, to make TMTO strategy work in these scenarios, we should calculate more $t_{\mathbf{a}}$ for different i's and search by bucket with round-up.

The precise description of TMTO strategy attack is below:

(1) $\forall a \in C_{\frac{\alpha}{2}}$, calculate the $t^i_{\mathbf{a}} = \sum a_j m_{ij}$ for chosen i's;
(2) search $t^i_{\mathbf{a}}$ by bucket search or other search algorithms;
(3) find all different codewords $(\mathbf{a}, \mathbf{b})$ that satisfies $t^i_{\mathbf{a}} = t^i_{\mathbf{b}}$ for all chosen i's, check whether $t^i_{\mathbf{a}} = t^i_{\mathbf{b}} + x, \forall 1 \le i \le n$ where $x = -\frac{\sqrt{2}}{\sqrt{\alpha}}, 0, \frac{\sqrt{2}}{\sqrt{\alpha}}$ and output $\frac{1}{\sqrt{2}}(\mathbf{a} - \mathbf{b})$ as codeword $\mathbf{c}_1$.

Complexity The probability that the equation $t^i_{\mathbf{a}} = t^i_{\mathbf{b}}$ is correct for random i is $\frac{n-\alpha}{n}$. The number of codewords $C_{\frac{\alpha}{2}}$ is $2^{\frac{\alpha}{2}}\binom{n}{\frac{\alpha}{2}}$. For each codeword in $C_{\frac{\alpha}{2}}$, it needs $\frac{\alpha}{2}$ additions to compute t^i. Thus, with success expectation of 1, we need total $\frac{\alpha n}{2(n-\alpha)}|C_{\frac{\alpha}{2}}|$ additions and also need memory to store $|C_{\frac{\alpha}{2}}|$ codewords with t^i. If the step 1-3 can be done together, we can early terminate the calculation of step 2 if satisfactory pairs of codewords have found. As for codeword $\mathbf{c}_1 \in C_\alpha$, there are $\binom{\alpha}{\frac{\alpha}{2}}$ pairs in $C_{\frac{\alpha}{2}}$ that sums to $\sqrt{2}\mathbf{c}_1$. Therefore, the storage requirement and number of additions can be decreased by factor $\sqrt{\binom{\alpha}{\frac{\alpha}{2}}}$.

Here we assume that there are few pairs satisfying the conditions in step 3. However, since the precision of t^i is limited, the satisfying pairs might be too large, making the computation cost of checking each pairs on other entries unacceptable. For example, if t^i is 32-bit format float and distributed in range $(-1, 1)$, it can only exclude magnitude of 2^{33} pairs of codewords. Thus, to exclude enough pairs of codewords, we need to calculate t^i for more different i's. It will slightly enlarge the storage requirement with factor $\#i$(number of chosen i's) and number of additions with factor $\#i * (\frac{n}{n-\alpha})^{\#i}$. If the the equation 11 contains some noise, the required number of i's would be more.

We ignore the complexity of search algorithm. As if the search algorithm is bucket search, the time and storage complexities are both $O(|C_{\frac{\alpha}{2}}|)$, comparable to the complexity of additions.

For concrete settings as $n = 512, \alpha = 16$, the requirement of storage is around $2^{57.8}$ codewords and each codeword needs $8 * \log_2(512) = 72$ bit $= 9$ bytes with at least 4 bytes for storage of t^i, which makes the storage larger than 2.8 EB. And the requirement of additions is around $2^{60.8}$.

B Remark on Limited Space of Secure Sketch

Here we give an attack if the sketch is generated as in Section 6.1, i.e. $\mathrm{SS}(\mathbf{w}) = \mathbf{T}\mathbf{M}_1$ where $\mathbf{M}_1$ is naive rotation matrix from $\mathbf{w}$ to

predefined fixed codeword $\mathbf{c}_{fixed}$ and $\mathbf{T}$ is defined in Theorem 6.1. First, we give the proof of the Theorem 6.1.

Theorem 6.1. Assume $\mathbf{T} = (t_{ij})$, $\mathbf{a} = (a_1, \ldots, a_n) \in C_\alpha$ where $a_j = \frac{1}{\sqrt{\alpha}}$. Then $\mathbf{a}' = (a'_1, \ldots, a'_n) \in C_\alpha$ where $\forall k \neq j$, $a'_k = a_k$ and $a'_j = -a_j$. Because of the definition of $\mathbf{T}$, $\mathbf{Ta}, \mathbf{Ta}' \in C_\alpha$. We have

$$\mathbf{Ta} - \mathbf{Ta}' = \mathbf{c} - \mathbf{c}' \tag{14}$$

$$\frac{2}{\sqrt{\alpha}} t_{ij} = \pm\frac{2}{\sqrt{\alpha}}, \pm\frac{1}{\sqrt{\alpha}}, 0 \tag{15}$$

$$t_{ij} = \pm 1, \pm 0.5, 0 \tag{16}$$

$\forall i, j \in [n]$. Since $\mathbf{T}$ is an orthogonal matrix, the norm of each row of $\mathbf{T}$ is 1. There are two cases for each row of $\mathbf{T}$. One is that it consists of four positions filled with ± 0.5, the other is that it only has one position filled with ± 1. Here we prove that the first case is unsatisfactory by showing that if exist row i of $\mathbf{T}$ satisfies first case, $\exists \mathbf{c} \in C_\alpha$, $\mathbf{Tc} \notin C_\alpha$.

For row i of $\mathbf{T}$, assume $t_{ij_1}, t_{ij_2}, t_{ij_3}, t_{ij_4} = \pm 0.5$. If $\alpha = 1$, $\mathbf{t}_{i*} * \mathbf{c} = 0.5$ with $c_{j_1} = 1$. Then $\mathbf{Tc} \notin C_\alpha$. If $\alpha \geq 3$, construct $\mathbf{c} \in C_\alpha$ so that $\forall 1 \leq k \leq 3, c_{j_k} = sign(t_{ij_k}) \frac{1}{\sqrt{\alpha}}$ and $c_{j_4} = 0$. Then $\mathbf{t}_{i*} * \mathbf{c} = \frac{3}{2\sqrt{\alpha}}$ and $\mathbf{Tc} \notin C_\alpha$.

As each row of $\mathbf{T}$ is equal to $\pm\mathbf{e}_i^T$ and $\mathbf{T}$ is full of rank, the row vectors of $\mathbf{T}$ can be seen as a permutation of $\mathbf{e}_0^T, \mathbf{e}_1^T, \cdots, \mathbf{e}_n^T$. The column vectors of $\mathbf{T}$ are similar. So $\mathbf{T}$ can be written as

$$\begin{pmatrix} \pm\mathbf{e}_{i_1} & \pm\mathbf{e}_{i_2} & \cdots & \pm\mathbf{e}_{i_n} \end{pmatrix} \tag{17}$$

where $\mathbf{e}_{i_1}, \mathbf{e}_{i_2}, \cdots, \mathbf{e}_{i_n}$ is a permutation of unit vectors $\mathbf{e}_0, \mathbf{e}_1, \cdots, \mathbf{e}_n$. □

Then we recall the naive isometry rotation defined in [26].

Definition B.1 (naive isometry rotation). Given vectors $\mathbf{t}, \mathbf{c} \in \mathbb{R}^n$, let $\mathbf{w} = \mathbf{c} - \mathbf{t}^T\mathbf{ct}$ and $\mathbf{R}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ where $\theta = Angle(\mathbf{t}, \mathbf{c})$. The naive rotation matrix $\mathbf{R}$ mapping from $\mathbf{t}$ to $\mathbf{c}$ is:

$$\mathbf{R} = \mathbf{I} - \mathbf{tt}^T - \mathbf{ww}^T + \begin{pmatrix} \mathbf{t} & \mathbf{w} \end{pmatrix} \mathbf{R}_\theta \begin{pmatrix} \mathbf{t} & \mathbf{w} \end{pmatrix}^T \tag{18}$$

where $\mathbf{I}$ is identity matrix.

The naive isometry rotation $\mathbf{R}$ can be seen as rotating $\mathbf{t}$ to $\mathbf{c}$ in the 2D plane P extended by $\mathbf{t}$ and $\mathbf{c}$. Therefore, there are vectors in $(n-2)$ dimension subspace orthogonal to P that satisfy $\mathbf{Rv} = \mathbf{v}$. Restrict that $\mathbf{v}$ only has few non-zero positions, we have $\mathbf{TRv} = \mathbf{Tv}$ filled with lots of 0s. By guessing the non-zero positions in $\mathbf{v}$ and zero positions in $\mathbf{TRv}$, we can calculate and filter to get $\mathbf{v}$. We can calculate the null space P' of $\mathbf{M}'$ made of these vectors. As $\mathbf{v} \perp P$, with enough vectors, we can greatly shrink the space P' to P and finally retrieve P. If $\mathbf{t}$ is the biometric template and $\mathbf{c} \in C_\alpha$, it's easy to retrieve $\mathbf{t}$ knowing plane P.

However, by experiments, we find that the first null vector $\mathbf{v}'$ of $\mathbf{M}'$ is close enough to biometric template $\mathbf{t}$. Hence, we just take $\mathbf{v}'$ as possible candidate and use the sketch algorithm to try to retrieve original template $\mathbf{t}$ or $-\mathbf{t}$. The details are shown in Algorithm 5.

With $n = 512$, $\alpha = 16$, Algorithm 5 can output original template $\mathbf{w}$ or $-\mathbf{w}$ with probability $\approx 60\%$ on 200 tests by setting $m = 8$ and $\theta_t = 30°$.

Algorithm 5: Template Retrieve Attack on $\mathbf{M} = \mathbf{TR}$

Data: $\mathbf{M} = \mathbf{TR} \in O(n)$ with $\mathbf{T} \in \mathcal{T}$ and $\mathbf{R}$ is the native isometry rotation, threshold θ_t, m
Result: $\mathbf{v}$ or $\perp$
Create empty set V
for $k = 2, \ldots, m$ **do**
  for $i' = 1, \ldots, n$ `// repeat n times` **do**
    Random select distinct k indices $I = (i_1 < \ldots < i_k \in [n])$
    Random select distinct k indices $J = (j_1 < \ldots < j_k \in [n])$
    Select the submatrix $\mathbf{M}'$ of $\mathbf{M}$ with row indices I and column indices J
    Compute the null vector $\mathbf{u} = (u_1, \ldots, u_k)^T$ of submatrix $\mathbf{M}'$
    Compute $\mathbf{v} = (v_1, \ldots, v_n)^T$ such that $v_{j_i} = u_i, \forall j_i \in J$ and $v_j = 0, \forall j \notin J$
    if $\mathbf{Mv}$ *contains exactly* i *non-zero positions* `// make v satisfy Rv = v` **then**
      Store $\mathbf{v}$ in V
$\mathbf{M}' \leftarrow$ vertical stack of vectors in V
$V' \leftarrow$ approximate null vectors of $\mathbf{M}'$
for $\mathbf{v} \in V'$ **do**
  $\mathbf{c}' \leftarrow \mathbf{Decode}(\mathbf{Mv})$
  if $Angle(\mathbf{Mv}, \mathbf{c}') < \theta_t$ **then**
    Output $\mathbf{v}$
Output $\perp$

C Reduction of Threat Model

Theorem C.1. *For the hypersphere secure sketch* $SS = (\mathbf{SS}, \mathbf{Rec})$ *defined in Definition 3.5, if there exists an attacker* $\mathcal{A}$ *such that* $\Pr[\mathrm{SSMUL}_{\mathcal{A},\theta,t,\theta'}(n) = 1]$ *is non-negligible, provided that* θ *and* θ' *satisfy* $\theta \leq \theta'$ *and* $\forall \theta'' \leq \theta'$, $\Pr[\mathbf{Rec}(\mathbf{w}', \mathbf{SS}(\mathbf{w})) = \mathbf{w} | \mathbf{w}^T\mathbf{w}' = \cos\theta'']$ *is non-negligible, then there exists an attacker* $\mathcal{A}'$ *such that* $\Pr[\mathrm{SSMUL}_{\mathcal{A}',\theta,t}(n) = 1]$ *is also non-negligible.*

Proof. The attacker $\mathcal{A}'$ in experiment $\mathrm{SSMUL}_{\mathcal{A}',\theta,t-1}(n)$ do as follows:

(1) Get the protected template $\mathbf{SS}(\mathbf{w})$ from the challenger C, and send it to attacker $\mathcal{A}$;
(2) Receive a query $q \leq t$ from attacker $\mathcal{A}$. Get the set of protected templates $Q = \{\mathbf{M}_i \in \mathbf{SS}(\mathbf{w}_i), i \in [q]\}$ from challenger C, and send Q to the attacker $\mathcal{A}$;
(3) Receive the output $\mathbf{w}'$ from $\mathcal{A}$, and output $\mathbf{Rec}(\mathbf{w}', \mathbf{SS}(\mathbf{w}))$.

The attacker $\mathcal{A}'$ simulates the experiment $\mathrm{SSMUL}_{\mathcal{A},\theta,t,\theta'}(n)$ for the attacker $\mathcal{A}$. Therefore, we have:

$$\Pr[\mathrm{SSMUL}_{\mathcal{A}',\theta,t}(n) = 1] \geq \Pr[\mathrm{SSMUL}_{\mathcal{A},\theta,t,\theta'}(n) = 1] \times \Pr[\mathbf{Rec}(\mathbf{w}', \mathbf{SS}(\mathbf{w})) = \mathbf{w} | \mathbf{w}'^T\mathbf{w} \geq \cos\theta']$$

Thus, $\Pr[\mathrm{SSMUL}_{\mathcal{A}',\theta,t-1}(n) = 1]$ is non-negligible. □

By experiments,we found that the hypersphere sketch with error correcting codes as defined in Definition 3.2 satisfies $\forall\theta'' \leq \theta, \Pr[\mathbf{Rec}(\mathbf{w}', \mathbf{SS}(\mathbf{w})) = \mathbf{w} | dis(\mathbf{w}', \mathbf{w}) = \theta''] \geq 42.9\%$ considering the randomness of $\mathbf{SS}$ when $\theta = 49°$ in dimension $n = 512$ and $alpha = 16$. This threshold θ is larger than the criterion of IronMask, which is 37°. Therefore, we conclude that the conclusion of Theorem C.1 holds.

D Proof of Theorem 6.2

Proof. We begin by expressing $\cos\theta_{24}$ as follows:

$$\begin{aligned}\cos\theta_{24} = \mathbf{w}_2^T\mathbf{w}_4 &= (\cos\theta_{12}\mathbf{w}_1^T + \sin\theta_{12}\mathbf{v}^T)(\cos\theta_{34}\mathbf{w}_3 + \sin\theta_{34}\mathbf{u}) \\ &= \cos\theta_{12}\cos\theta_{13}\cos\theta_{34} + \sin\theta_{12}\cos\theta_{34}\mathbf{v}^T\mathbf{w}_3 \\ &+ \sin\theta_{34}\cos\theta_{12}\mathbf{w}_1^T\mathbf{u} + \sin\theta_{12}\sin\theta_{34}\mathbf{v}^T\mathbf{u}.\end{aligned}$$

Next, we decompose $\mathbf{w}_3 = \cos\theta_{13}\mathbf{w}_1 + \sin\theta_{13}\mathbf{r}$, where r is a unit vector orthogonal to $\mathbf{w}_1$. Substituting this into the expression for $\mathbf{v}^T\mathbf{w}_3$, we have:

$$\mathbf{v}^T\mathbf{w}_3 = \mathbf{v}^T(\cos\theta_{13}\mathbf{w}_1 + \sin\theta_{13}) = \sin\theta_{13}\mathbf{v}^T\mathbf{r}.$$

Since $\mathbf{v}$ is a random unit vector orthogonal to $\mathbf{w}_1$ and $\mathbf{r}$ is also orthogonal to $\mathbf{w}_1$, the expectation of $\mathbf{v}^T\mathbf{w}_3$ is zero:

$$E(\mathbf{v}^T\mathbf{w}_3) = E(\sin\theta_{13}\mathbf{v}^T\mathbf{r}) = 0.$$

Similarly, we have $E(\mathbf{w}_1^T\mathbf{u}) = 0$. Combining these results, the expectation of $\cos\theta_{24}$ simplifies to:

$$\begin{aligned}E(\cos\theta_{24}) &= E(\cos\theta_{12}\cos\theta_{13}\cos\theta_{34} + \sin\theta_{12}\sin\theta_{34}\mathbf{u}^T\mathbf{v}) \\ &= \cos\theta_{12}\cos\theta_{13}\cos\theta_{34}.\end{aligned}$$

This completes the proof. □