Privacy-Preserving Identity Management and Software Bill of Materials Vulnerability Detection: Practical Use Cases from the PRIVIDEMA Project

Mariya Georgieva Belorgey[1], Benoit Cogliati[2], Simon Demarty[2,3] Lois Huguenin-Dumittan[1], Özcan Ozturk[4], Salma Samiei[5], and Oana Stan[6]

[1] TuneInsight
[2] Thales DIS France SAS, Meudon, France
[3] Laboratoire de Mathématiques de Versailles, UVSQ, CNRS, Université Paris-Saclay, Versailles, France
[4] Sabanci University
[5] Idakto
[6] Université Paris-Saclay, CEA LIST, 91120, Palaiseau, France

Abstract. PRIVIDEMA project (*Privacy-Preserving Identity Management for Digital Wallets and Secure Data Sharing and Processing for Cyber Threat Intelligence Data*) advances the state of the art in cryptographic and Privacy-Enhancing Technologies (PETs) to enable secure, interoperable, and trustworthy data exchange across sectors, with a focus on the domains of **Cyber Threat Intelligence** and **Digital Identity Management**. This paper presents two representative real-world use cases: **(1)** privacy-preserving digital identity management based on the European Digital Identity (EUDI) Wallet, and **(2)** privacy-preserving Cyber Threat Intelligence (CTI) sharing for Software Bill of Materials (SBOMs) and vulnerability datasets. Both use cases showcase how advanced PETs, including Fully Homomorphic Encryption (FHE), Federated Learning (FL), and Differential Privacy (DP), can be composed to protect sensitive data throughout its lifecycle while maintaining analytical and operational utility. Together, these use cases chart a practical course toward more scalable, standards-compliant, and privacy-preserving data ecosystems that align with Europe's vision for secure and trustworthy digital services.

Keywords: Cyber-Threat Intelligence · Digital Identity Management · PET · Homomorphic Encryption.

1 Introduction and Context

The growing exchange of sensitive data, including personal identity attributes, biometric information, and cyber threat intelligence, across digital services, demands stronger guarantees for privacy, confidentiality, and regulatory compliance throughout the entire data lifecycle. Conventional protection mechanisms such

as encryption in transit, encryption at rest, and anonymisation provide important safeguards, but they remain insufficient for modern distributed ecosystems in which data must also be processed, analysed, and shared among multiple stakeholders. In most existing architectures, this still requires decryption prior to computation, creating exposure points that are particularly problematic in high-assurance domains.

This challenge has become strategically important in the European context, where secure cross-border interoperability and trustworthy digital services are central policy objectives. Emerging initiatives in digital identity and collaborative cybersecurity increasingly depend on infrastructures capable of preserving confidentiality not only during storage and communication, but also during authentication, analytics, and cooperative decision-making. As a result, there is a growing need for practical Privacy-Enhancing Technologies (PETs) that support secure computation directly over protected data.

The PRIVIDEMA project, funded under Horizon Europe, addresses this need through the development of a modular and scalable privacy-preserving data sharing framework. Its architecture combines advanced PETs, including Fully Homomorphic Encryption (FHE), Federated Learning (FL), and Differential Privacy (DP), to ensure that sensitive data remains protected even during processing. By integrating these complementary techniques, the project aims to bridge the gap between state-of-the-art privacy-preserving cryptography and deployable solutions that satisfy real-world operational, performance, and interoperability constraints.

The motivation for this work lies in the absence of practical mechanisms that allow organisations and users to benefit from remote services without disclosing the sensitive data required to operate them. This gap is particularly visible in scenarios where authentication decisions or security analytics depend on highly confidential inputs that cannot be safely revealed to third-party infrastructures. Addressing this limitation is essential for enabling trustworthy digital services that remain aligned with Europe's principles of privacy by design, digital sovereignty, and secure collaboration.

To address these challenges, PRIVIDEMA develops and validates two representative real-world use cases:

- **Use Case 1:** Privacy-Preserving Digital Identity (EUDI Wallet), this use-case explores how secure Multi-Factor Authentication (MFA) can be enhanced through the integration of FHE-based biometric authentication, providing robust identity verification while preserving user privacy, even during remote or untrusted computations.
- **Use Case 2:** Software Bill of Materials Vulnerability Detection. A key way to assess system security is by using a Software Bill of Materials (SBOM). This SBOM assists in understanding what components are inside a software. One role of such an inventory is to help people identify all the possible vulnerabilities that could impact the said software. These vulnerabilities are generally tracked against public databases such as MITRE's Common Vulnerabilities and Exposures database [12] (U.S.) or the European Union

Vulnerability Database [9]. Most tools that assess for vulnerabilities, however, work offline and only help individual companies protect their own software. Using privacy-preserving technologies like homomorphic encryption allows stakeholders to match Software Bills of Materials with vulnerability databases.

These two promising real-world use cases provide a strong foundation for developing a safer digital Europe, while ensuring the protection of biometric identity privacy in full alignment with EU policy.

2 Use Case 1 – Privacy-Preserving Digital Identity (EUDI Wallet)

Europe is building a unified digital identity infrastructure through the EU Digital Identity (EUDI) Wallet, proposed by the European Commission under the eIDAS 2.0 regulation [2]. The EUDI Wallet enables interoperable digital identity across EU member states, involving three core parties: the *identity provider* (issues identity attributes), *relying parties* (services requiring identity verification), and the *holder* (the citizen). The Architecture and Reference Framework (ARF) [14] standardizes identity issuance and presentation protocols, relying on ISO 18013-5 and SD-JWT containers to cryptographically bind identity attributes to their legitimate holder through user binding information, typically a public key tied to user-controlled private credentials.

To achieve the required Level of Assurance (LoA), the ARF supports secure hardware components such as Secure Elements (SE), Trusted Execution Environments (TEE), or remote Hardware Security Modules (HSM). While biometric authentication is supported locally via SE or TEE, **biometric verification is currently absent from the EUDI Wallet ecosystem in remote cloud settings**. This gap stems from fundamental privacy tensions: delegating biometric verification to a remote server exposes sensitive personal data to an infrastructure the user does not control, risks activity tracking across sessions, and conflicts with the self-sovereign identity principle at the core of the EUDI vision. Existing standards do not provide mechanisms to perform remote biometric matching without disclosing biometric templates or behavioral patterns to the server, making cloud-based biometric unlock of HSM-protected keys an open and unaddressed problem.

In this project, we aim to fill this gap by integrating *Fully Homomorphic Encryption* (FHE) into the remote biometric authentication pipeline for the EUDI Wallet. FHE enables computation directly on encrypted data, allowing a remote authentication server to perform biometric matching without ever accessing plaintext biometric features, while the HSM retains exclusive control over decryption and key-release decisions. Concretely, we target the following objectives:

- Enable HSM key unlocking conditioned on remote biometric verification, supporting 1:1 *authentication* (query embedding matched against a single enrolled template), while keeping all biometric data encrypted throughout.

– Guarantee that a curious or compromised server learns nothing about the user's biometric characteristics, and cannot link authentication events across sessions.
– Demonstrate that the performance overhead introduced by FHE remains acceptable within the operational constraints of the EUDI Wallet use case.

The proposed solution aligns with both the zero-trust security model and the self-sovereign identity principle: the server performs its function, verifying identity, without acquiring the data that makes that verification possible in plaintext.

2.1 Face Recognition Framework

The biometric authentication pipeline is built on InsightFace [11], an open-source framework for deep face analysis, using ArcFace [8] as the core recognition model. ArcFace's additive angular margin loss produces highly discriminative 512-dimensional facial embeddings, making it well-suited for privacy-preserving matching under encryption. The pipeline is partitioned into two trust domains: an *on-device* domain responsible for all biometric capture and processing, and a *server-side* secure environment consisting of an authentication server that performs homomorphic similarity computation, and a remote HSM that holds the decryption key and performs threshold evaluation and key-release decisions. All biometric data is encrypted before leaving the device, and no plaintext biometric material is ever transmitted or processed outside it.

The pipeline proceeds through the following stages, illustrated in Figure 1:

– **Capture and preprocessing.** The user's facial image is acquired via mobile camera and normalized (resizing, color correction) to ensure consistent input to the downstream recognition stages.
– **Detection and alignment** Locates the face and aligns it against five landmarks (eyes, nose, mouth corners), standardizing orientation prior to embedding extraction by RetinaFace [7].
– **Embedding generation.** ArcFace produces a compact 512-dimensional feature vector encoding the user's facial characteristics. This embedding is generated entirely *on the client device* and never transmitted in plaintext. The same procedure is applied both to the reference template, enrolled once and stored encrypted on the server, and to the fresh query image presented at each authentication attempt.
– **FHE encryption.** Both the query embedding and, during enrollment, the reference template embedding are encrypted on-device under the CKKS scheme, using a public key whose corresponding secret key is held exclusively by the HSM. Each encryption draws fresh randomness, ensuring that ciphertexts produced in distinct sessions are computationally indistinguishable. The resulting ciphertexts are then transmitted to the authentication server.

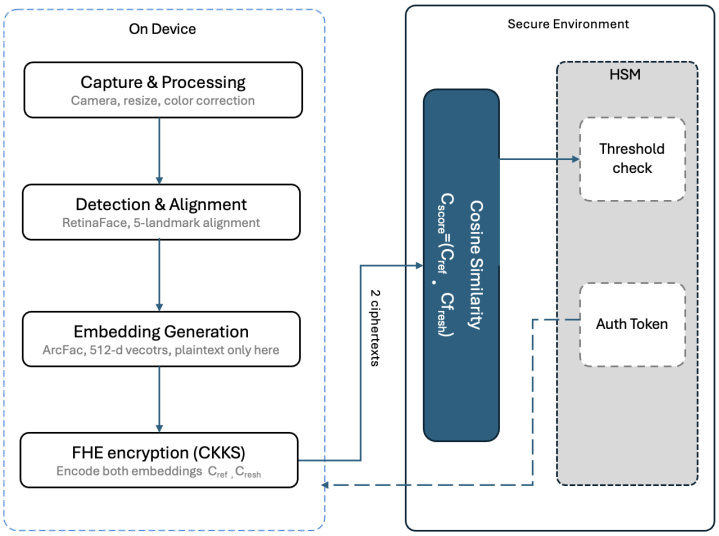

Fig. 1: Privacy-preserving biometric authentication pipeline for the EUDI Wallet.

– **Homomorphic matching.** The authentication server evaluates the cosine similarity between the encrypted query embedding and the encrypted stored template directly in ciphertext space, via an inner product over 512 CKKS-encoded coefficients. The resulting encrypted similarity score is forwarded to the HSM without decryption. The HSM decrypts the score within its protected boundary, compares it against the authentication threshold, and releases the protected key material only upon a successful match. No plaintext biometric data is accessible at any point outside the client device.

The face detection, alignment, and embedding extraction stages are executed entirely on the client device, where plaintext computation is both permissible and efficient. RetinaFace performs face localisation and five-landmark alignment on the raw image, after which ArcFace extracts the 512-dimensional embedding. Since these stages operate on plaintext data that never leaves the device, they incur no FHE overhead whatsoever. The sole stage requiring homomorphic evaluation is template matching: computing the cosine similarity between the encrypted query embedding and the encrypted template. This reduces to an encrypted inner product over 512 CKKS coefficients, implemented via the `prividema-fhe` library, a C++ implementation building on Lattigo [1]. The computation requires modest multiplicative depth and preserves matching accuracy within the approximation bounds of CKKS. This targeted use of FHE, applied only at the point where encrypted data must be compared on an untrusted server, avoids the prohibitive cost of evaluating CNN non-linearities homomorphically, which would otherwise constitute an intractable performance bottleneck.

Performance considerations. Performance is a critical dimension of the proposed system, as the authentication sub-flow is directly embedded in user-facing wallet operations, including credential issuance and presentation under the VCI and VP workflows. An end-to-end latency budget must therefore be defined and validated against the operational requirements of the EUDI Wallet. This budget must be evaluated holistically across all phases of the pipeline: on-device biometric capture and preprocessing, FHE encryption, network transport to the authentication server, homomorphic matching, encrypted score transfer to the HSM, and HSM-side decryption and threshold evaluation. Meeting this target across the full pipeline, rather than optimizing individual stages in isolation, is a key research and engineering objective of this work.

2.2 Threat Model

We consider the remote authentication server as an *honest-but-curious* adversary: it executes the biometric matching protocol correctly but attempts to infer information about the user's biometric characteristics or activity patterns from the data it receives. Additionally, we consider a passive network attacker capable of intercepting all communications between the client device and the server. Compromised storage is also considered, where an attacker gains read access to the server-side encrypted template database.

Under these assumptions, our solution provides the following guarantees.

Template confidentiality: the authentication server never accesses plaintext biometric embeddings. All data is encrypted under the CKKS scheme before transmission, ensuring that neither a curious server nor a storage attacker can recover facial features from the ciphertexts it holds.

Unlinkability: each authentication session produces a fresh ciphertext under independent randomness. Under the semantic security of the CKKS scheme, successive sessions are computationally indistinguishable to the server, preventing it from correlating authentication events or tracking user activity over time.

These properties directly fulfill the obligations from EU Regulation 2024/1183 (eIDAS 2.0) [2], specifically Article 5a(5), which prohibits wallet providers from obtaining any information about the use of electronic attestations, and Article 5a(14), which requires personal wallet data to remain logically separate from any other data held by the provider.

3 Use Case 2 – Privacy-Preserving Cyber Threat Intelligence Sharing

A Software Bill of Materials (SBOM) is an inventory of all components that have been used to build software. This includes all elements that are distributed as part of the final artifact, but also those that were used to develop and build it. One role of such an inventory is to help people identify all the possible vulnerabilities that would impact the said software. Such vulnerability detection checks whether the software of the SBOM includes known Common Vulnerabilities and

Exposures (CVEs) reported in the MITRE database [12] or vulnerabilities published in the European Union Vulnerability Database [9]. However, a large proportion of the vulnerability detection tools are designed to be used offline as they assess the risk for one's product. The presented use case addresses this issue and shows a centralised yet privacy-preserving tool. The core of this Use Case is to build a privacy-preserving centralised SBOM analysis service that can also offer advanced features to evaluate the potential impact of new CVEs on its users.

3.1 SBOM Framework

The use case is split into several steps.

- Cyber Threat Intelligence Collection: The users collect and continuously update their SBOMs, the CVEs that affect them, along with relevant data about their interaction with the service provider.
- Data Ingestion and Preprocessing: The service provider continuously collects new CVEs.
- Secure Model Training: A machine learning model, shared among several users, is trained in a distributed manner across the participating users, leveraging FL techniques to preserve individual data privacy.
- Privacy-Preserving Threat Detection: The CVE database is used to analyze software components and potential and confirmed vulnerabilities.
- Early Warning and Threat Sharing: The detected vulnerabilities are securely shared with users promptly, enabling proactive remediation and threat mitigation.

To address those steps, PRIVIDEMA proposes four different scenarios tackling one subset of the project each. The first and second scenarios let the service provider send to the users detailed information about impacted elements of their SBOMs, either through one-time query or continuously by monitoring stored encrypted SBOMs. Third scenario helps the service provider assess an approximation of the percentage of its impacted users through Differential Privacy and Threshold homomorphic encryption. Finally, users collaboratively build an anomaly detection model to monitor the service provider's behaviour and detect any deviation from its supposedly normal demeanour, thanks to federated learning.

In this paper, we focus below on the first scenario and give details on its structure and the size and quantity of involved data.

3.2 Data Consideration

In this use case, two types of entities will be interacting:

- a centralised entity which provisions a service of SBOM analyser. It is thus named service provider.
- a flock of entities, called users, that all have a SBOM which is to be analysed by the service provider.

Before continuing, it is important to discuss the scale of manipulated data. Indeed, the service provider does use a large amount of CVEs stored in its internal database. According to [12], there are more than 300,000 known CVEs to this day. Also, the amount of discovered CVEs is rapidly growing, reaching 48,000 new ones in 2025. To ease the transfer of data between the service provider and its users, the main pieces of information of each CVE are encoded in a small bitstring of 72-bits. Compared to non-encoded CVEs (often stored in JSON that amount for a total of tens of thousands of bits), those smaller values facilitate the application of the OnionPIR protocol [13].

Finally, the way we designed this use case induces data duplication in the CVEs database, as each row corresponds to a version of a software and contains a set amount N of CVE that affect it. Thus each CVE may appear for two versions of the same software. Hence the database size is equal to the multiplication of the size of an encoded CVE, the variable N, and the number of rows. Taking $N = 1000$ (which allows each row to contain 1000 CVEs), we can generously upper bound the size of the database by 900 Mbytes, which corresponds to at most 100 000 rows of 9000 bytes each.

3.3 Scenario

The main scenario of this use case brings a solution to the base problem we mentioned earlier: finding a way to detect vulnerabilities in a manufacturer's software artifact by matching its SBOM with a database of CVEs. The flow, which is represented in Figure 2, Figure 3 and Figure 4 is twofold. First, at the service provider level, some operations are run regularly:

- The service provider SP updates its database of software it monitors (cf. Figure 2) .
- The service provider SP updates its CVEs database (cf. Figure 3) .

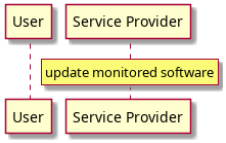

Fig. 2: Scenario 1: softwares collection.

Then, for a user to employ the service provided by SP, both SP and the user follows the protocol shown in Figure 4 and detailed below:

- The user asks the service provider for the encoding format needed for its SBOM.
- The user encodes its SBOM.

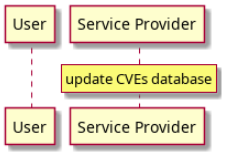

Fig. 3: Scenario 1: CVEs collection and preprocessing.

- The user encrypts its SBOM using its fully homomorphic encryption (FHE) public key.
- The user sends a query containing its SBOM and the public key to the service provider.
- The service provider runs the private information retrieval (PIR) protocol on its up-to-date database against the query in the encrypted domain.
- The service provider sends the answer back to the user.
- The user decrypts and decodes the answer.

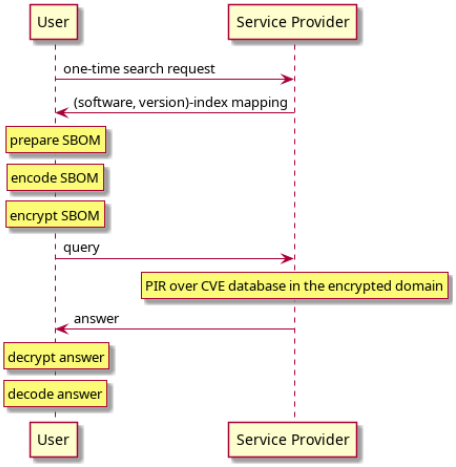

Fig. 4: Scenario 1: runtime operations.

To give a general idea of the PIR algorithm we use in this scenario and in the following ones, we consider that the service provider holds a database of CVEs as follows: each row index correspond to a pair (software, version of the software) and contains a fixed number of CVEs arbitrarily chosen (for instance the most critical ones). The PIR protocol evaluated in FHE essentially is a dot product between the database which is vectorised and the encrypted vectorised SBOM of the user. This is possible because of two points:

- in FHE, dot products are possible,

– the vectorised database of CVEs and the vectorised SBOM have the same format.

Indeed, the vectorised CVEs database basically is a vector where each element is a concatenation of encoded CVEs. When the user asks the service provider for the encoding format needed for its SBOM, it receives two pieces of information: the mapping (software, version)-index and the encoding table of CVEs. Thanks to the former, the user is able to produce a vector for its SBOM where each element is a zero except for the index corresponding to software that are in the SBOM. The vector elements at those indices are set to one. This is the vectorised SBOM that the user now encrypts and sends to the service provider. Finally, SP performs the PIR algorithm and sends the answer to the user, which uses the latter piece of information transmitted by the service provider to decode the result after decrypting it. It is possible to achieve better performance by compressing requests. One way of doing this is by representing data in a hypercube, as in the OnionPIR protocol [13].

Let us now describe more in details the homomorphic library build in PRIVIDEMA project which, together with Lattigo, will be used to implement both the biometric matching for use case 1 and the OnionPIR for use case 2.

4 Fully Homomorphic Encryption Library

The prividema-lib is the main unified FHE software library developed within the PRIVIDEMA project. It provides a common abstraction layer for multiple homomorphic encryption schemes, including BFV [10], CKKS [5] and TFHE/CGGI [6], through a shared GLWE/GGSW-based ciphertext representation. This unified design enables interoperability across schemes while preserving the strengths and optimizations specific to each of them.

A central feature of prividema-lib is its support for efficient scheme switching based on Chimera framework [4], allowing ciphertexts to transition from one scheme representation to another without requiring decryption. This capability enables hybrid encrypted computation workflows in which, for example, CKKS is used for approximate numerical processing, while TFHE is used for fast Boolean operations, comparisons, and control-flow logic.

The library is built on the base-2^k bivariate representation and relies on the spqlios-arithmetic [3] backend for high-performance FFT/NTT operations. This architecture significantly accelerates core primitives such as external products, key switching, automorphisms, relinearization, and bootstrapping-related transformations, offering substantial speedups compared with traditional full-RNS implementations.

At this stage of the project, prividema-lib almost implements the main cryptographic modules and arithmetic backends. The remaining work primarily focuses on scheme-switching interfaces and final optimization passes, hardware-specific acceleration, API stabilization, and integration with the higher-level privacy-preserving circuits required by the project use cases. Within the overall software stack, prividema-lib serves as the core cryptographic engine that bridges

low-level arithmetic modules and high-level privacy-preserving applications, including biometric authentication, secure aggregation, and privacy-preserving cyber threat intelligence workflows.

4.1 Performance of OnionPIR with prividema-lib

We evaluated the OnionPIR workflow implemented with prividema-lib using the standard OnionPIR parameter set with $N = 4096$, which is representative of the SBOM vulnerability-detection use case. Preliminary benchmarking results show a significant performance improvement compared with the original implementation [13].

For a 500 GB encrypted database, the end-to-end PIR processing time is reduced to approximately 2600s, compared with 6500s in the earlier baseline, corresponding to a speedup of roughly $\times 2.5$. This gain is primarily attributed to the optimized arithmetic backend and the more efficient ciphertext transformation primitives available in prividema-lib.

In terms of throughput, we observe a peak bandwidth between 120 and 200 MB/s on standard cloud machines, while compute-optimized instances reach between 140 and 240 MB/s. These results significantly improve upon the previous 80–100 MB/s range, demonstrating the practical benefit of the new implementation for large-scale vulnerability database queries.

These first results confirm that prividema-lib enables OnionPIR-style private retrieval at scales compatible with industrial SBOM analysis services, while preserving the privacy guarantees required by the PRIVIDEMA threat-intelligence use case.

5 Conclusion and future work

This paper presented the PRIVIDEMA project's vision for practical privacy-preserving data processing, illustrated through two representative use cases: remote biometric authentication for the EUDI Wallet and privacy-preserving SBOM vulnerability detection. These two scenarios demonstrate how advanced Privacy-Enhancing Technologies (PETs),notably Fully Homomorphic Encryption, can be used to support secure, interoperable, and regulation-aligned digital services. A key enabler of this vision is prividema-lib, the project's unified FHE software framework, which provides efficient support for BFV, CKKS, TFHE, and scheme switching within a single cryptographic stack.

Future work will focus on four main directions. First, we will complete the scheme-switching interfaces and performance optimization layers, including hardware-aware acceleration and API stabilization. Second, we will extend the biometric authentication pipeline toward fully encrypted face detection and alignment, which remains the main computational bottleneck. Third, the SBOM analysis service will be enhanced with richer privacy-preserving analytics, including early-warning mechanisms for newly disclosed CVEs and privacy-preserving collaborative learning across stakeholders. Finally, large-scale validation on real

deployment environments and cross-border interoperability scenarios will be carried out to assess scalability, usability, and compliance with the European digital identity and cybersecurity ecosystem.

Overall, PRIVIDEMA shows that advanced cryptographic techniques are moving from theoretical feasibility toward easily deployable, standards-compliant privacy-preserving infrastructures, contributing to Europe's broader objectives of digital sovereignty, privacy by design, and trustworthy data sharing ecosystems.

Disclosure of Interests. Funded by the European Union under Grant Agreement No. 101167964. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Cybersecurity Competence Centre. Neither the European Union nor the European Cybersecurity Competence Centre can be held responsible for them.
Project: PRIVIDEMA
Funding: Horizon Europe – Grant Agreement No. 101167964
Website: https://prividema.eu

Preprint Notice This document is a preprint of the following publication. The final authenticated version of this work has been published by *Springer* and is available at:

DOI: https://doi.org/10.1007/978-3-032-28957-5_19

Please cite the published version of record whenever possible. This preprint is made available in accordance with Springer Nature's self-archiving policy. The published version is the authoritative version.

References

1. Lattigo v6. Online: https://github.com/tuneinsight/lattigo, August 2024. EPFL-LDS, Tune Insight SA.
2. Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 april 2024 amending regulation (EU) no 910/2014 as regards establishing the european digital identity framework. Official Journal of the European Union, L 1183, 30 Apr 2024, 2024. Entered into force 20 May 2024 https://eur-lex.europa.eu/eli/reg/2024/1183/oj.
3. Mariya Georgieva Belorgey, Sergiu Carpov, Nicolas Gama, Sandra Guasch, and Dimitar Jetchev. Revisiting key decomposition techniques for FHE: Simpler, faster and more generic. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 176–207. Springer, 2024.
4. Christina Boura, Nicolas Gama, Mariya Georgieva, and Dimitar Jetchev. Chimera: Combining Ring-LWE-based fully homomorphic encryption schemes. *Journal of Mathematical Cryptology*, 14(1):316–338, 2020.
5. J. H. Cheon, A. Kim, M. Kim, and Y. Song. Homomorphic encryption for arithmetic of approximate numbers. In *ASIACRYPT*, pages 409–437.

6. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: Fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(2):34–91, 2020.
7. Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localization in the wild., 2019. https://arxiv.org/abs/1905.00641.
8. Deng, Jiankang, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
9. European Union Agency for Cybersecurity. European Union Vulnerability Database, 2025. Accessed: 2025-10-29.
10. J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. Technical Report 2012/144, IACR ePrint, 2012.
11. InsightFace. InsightFace: State-of-the-art 2d and 3d face analysis project, 2025. https://github.com/deepinsight/insightface.
12. MITRE Corporation. Common Vulnerabilities and Exposures (CVE), 2025. https://cve.mitre.org/ Accessed: 2025-10-29.
13. Muhammad Haris Mughees, Hao Chen, and Ling Ren. OnionPIR: Response efficient single-server PIR. In Yongdae Kim, Jong Kim, Giovanni Vigna, and Elaine Shi, editors, *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, pages 2292–2306. ACM, 2021.
14. European Digital Identity Wallet. Architecture and reference framework. Technical report, EUDI Wallet, April 2023.