Cryptographic Techniques

Encryption Fundamentals

Symmetric Encryption:

- Same key used for encryption and decryption
- Fast, efficient for large datasets
- Examples: AES, ChaCha20
- Challenge: Secure key distribution

Asymmetric Encryption:

- Public/private key pairs
- Slower but solves key distribution problems
- Examples: RSA, ECC
- Applications: Secure communications, digital signatures

Hybrid Systems:

- Use asymmetric for key exchange, symmetric for data encryption
- Examples: TLS, Signal protocol

Advanced Cryptographic Techniques

Secure Multi-party Computation (MPC):

- Allows multiple parties to jointly compute functions over inputs while keeping inputs private
- Applications: Privacy-preserving analytics, federated learning enhancement
- Implementation options: Garbled circuits, secret sharing

Zero-Knowledge Proofs:

- Prove knowledge of a value without revealing the value itself
- Applications: Authentication, compliance verification
- Types: zk-SNARKs, Bulletproofs

Searchable Encryption:

- Allows searching encrypted data without decryption
- Balances functionality and privacy
- Applications: Encrypted databases, secure cloud storage

Attribute-Based Encryption:

- Access control embedded into encryption
- Enables fine-grained data sharing policies
- Applications: Healthcare data sharing, IoT

Selection Criteria for Cryptographic Solutions

- Data volume and processing requirements
- Trust model (who can see what data)
- Regulatory requirements
- Performance constraints
- Required functionality (analysis, sharing, etc.)

Decision Framework Components

For Differential Privacy

Key Questions for Tool Implementation:

1. What type of data is being protected? (numerical, categorical, text)
2. What is the sensitivity level of the data? (low, medium, high)
3. Is there a trusted central authority? (yes/no)
4. What accuracy level is required? (low, medium, high)
5. How many queries are expected? (few, many, continuous)

Epsilon Selection Guidance:

- High sensitivity (medical): $\varepsilon = 0.1\text{-}1$
- Medium sensitivity (demographics): $\varepsilon = 1\text{-}3$
- Lower sensitivity (aggregated usage): $\varepsilon = 3\text{-}10$

For Cryptographic Techniques

Key Questions for Tool Implementation:

1. Who needs access to the raw data? (single party, multiple parties)
2. What operations need to be performed on the data? (storage, analysis, sharing)
3. What are the performance requirements? (real-time, batch processing)
4. What are the regulatory requirements? (GDPR, HIPAA, etc.)
5. What is the threat model? (external attackers, internal threats, etc.)

Integration Points With Other Privacy Techniques

- **Differential Privacy + Federated Learning**: Adding DP noise during federated learning training
- **Cryptography + Homomorphic Encryption**: Enhances MPC capabilities
- **Differential Privacy + Anonymization**: Adding DP as a post-processing step after anonymization
- **Cryptographic Techniques + Legal Frameworks**: Implementing encryption requirements of regulations

Practical Implementation Considerations

Differential Privacy Libraries

- Google's Differential Privacy library
- OpenDP (Harvard)
- IBM's Diffprivlib
- Microsoft's SmartNoise

Cryptography Implementation

- OpenSSL
- libsodium
- Microsoft SEAL (for homomorphic encryption)
- TFHE (for homomorphic encryption)

Trade-offs to Highlight in Your Tool

1. **Privacy vs. Utility**: Stronger privacy generally means less accurate results
2. **Complexity vs. Usability**: More sophisticated techniques can be harder to implement correctly
3. **Performance vs. Security**: Stronger security often requires more computational resources
4. **Centralized vs. Decentralized**: Trust requirements vs. computational efficiency