NTNU | Norwegian University of Science and Technology

Privacy-Preserving Cryptography from Zero-Knowledge Proofs

Ph.D. Defence, Tjerand Aga Silde, August 22nd, 2022

Content

Papers

Paper 1:	Anonymous Tokens with Public Metadata and Applications to Private Contact Tracing
Paper 2:	Lattice-Based Proof of Shuffle and Applications to Electronic Voting
Paper 3:	Verifiable Mix-Nets and Distributed Decryption for Voting from Lattice-Based Assumptions
Paper 4:	Verifiable Decryption in the Head
Paper 5:	Verifiable Decryption for BGV

Introduction

Many real-world systems require that users are authenticated or that information is certified while keeping the identity or content secret.

Some recent popular examples are anonymous browsing with spam-protection, anonymous telemetry collection, privacy-preserving contact-tracing, anonymous broadcasting, outsourced computation and electronic voting.

Introduction

Our main building block: zero-knowledge proofs.

1. A prover holds a secret witness w to some statement x
2. He wants to convince a verifier about w without revealing it
3. The prover and verifier interacts to convince the verifier
4. Correctness: if prover knows w then the verifier accepts
5. Soundness: if prover does not know w then the verifier rejects
6. Zero-Knowledge: verifier learns nothing about w but x is true

Introduction

The security of public-key cryptosystems is mostly based on hard computational problems: factoring large bi-primes or computing discrete logarithms over finite fields or elliptic curve groups.

Shor developed an algorithm that, if implemented on a large quantum computer, would efficiently solve these problems. This means that we need to design new cryptosystems that are secure against quantum computers.

Introduction

The main research goal of this thesis was to design new protocols based on zero-knowledge proofs for privacy applications. Four out of five papers in this thesis build systems that are quantum secure.

This thesis is based on joint work with Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, Thomas Haines, Johannes Muller, Peter Rønne, Martin Strand and Thor Tunge.

Anonymous Communication

Anonymous Tokens for Private Contact Tracing

The Norwegian Institute of Public Health has developed an app “Smittestopp” to supplement traditional contact tracing.

The app sends you a notification if you have been close to someone that has tested positive for Covid 19.

The hope is that this may be faster and may notify contacts that you forgot about or didn’t know about.

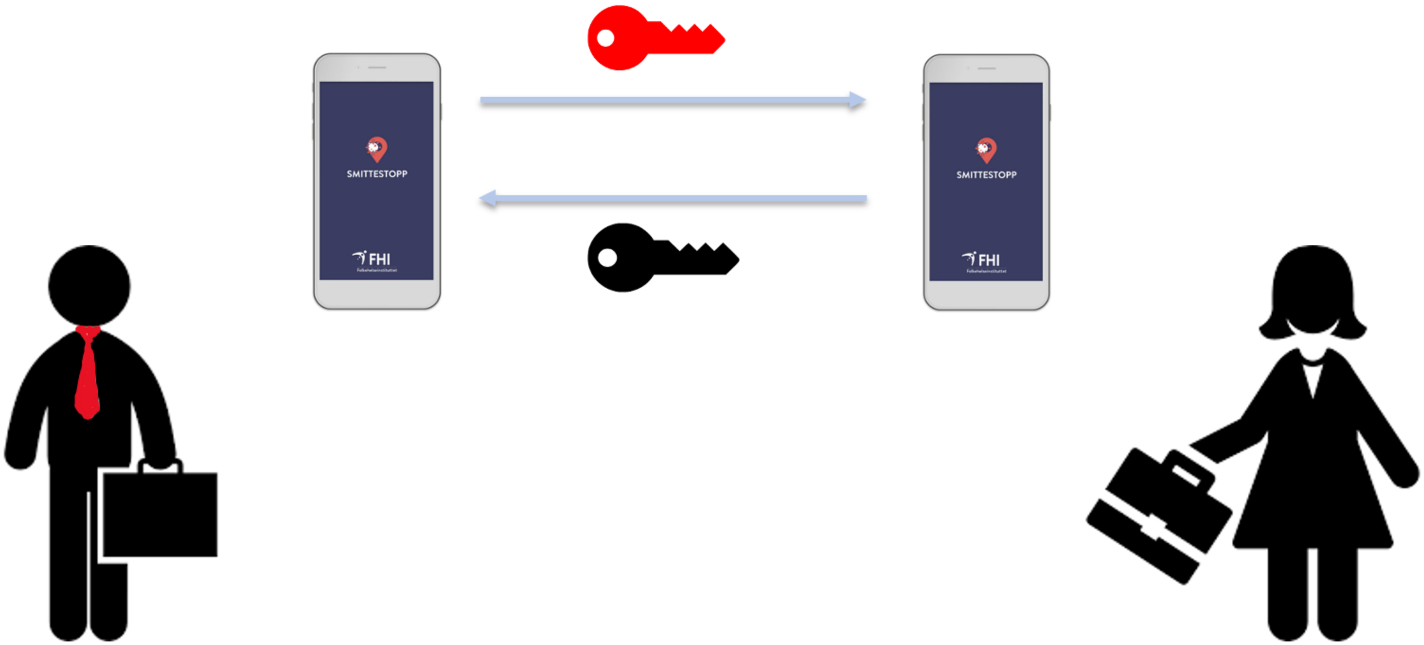
Anonymous Tokens for Private Contact Tracing

All data is stored on the user's phone. It uses Bluetooth for communication with other phones, but no GPS tracking.

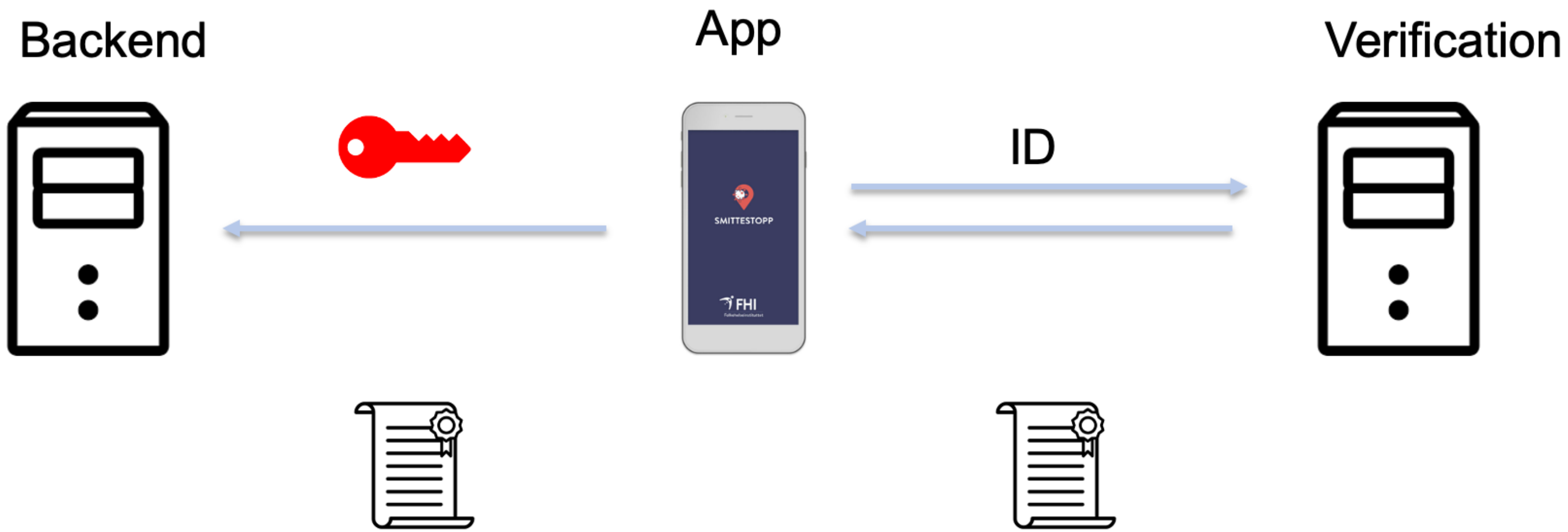
You only identify yourself to report a positive test, and then you upload the "infections keys" to the server.

The other users check locally if they have been in touch with someone who has uploaded their keys.

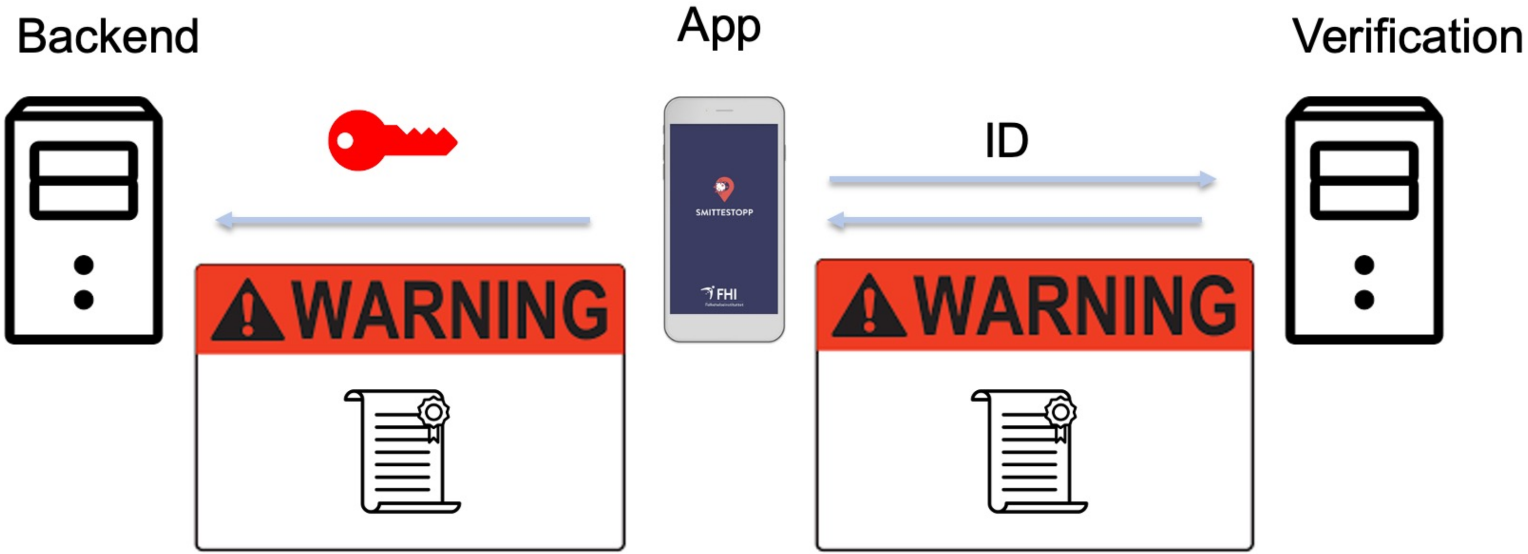
Anonymous Tokens for Private Contact Tracing



Anonymous Tokens for Private Contact Tracing

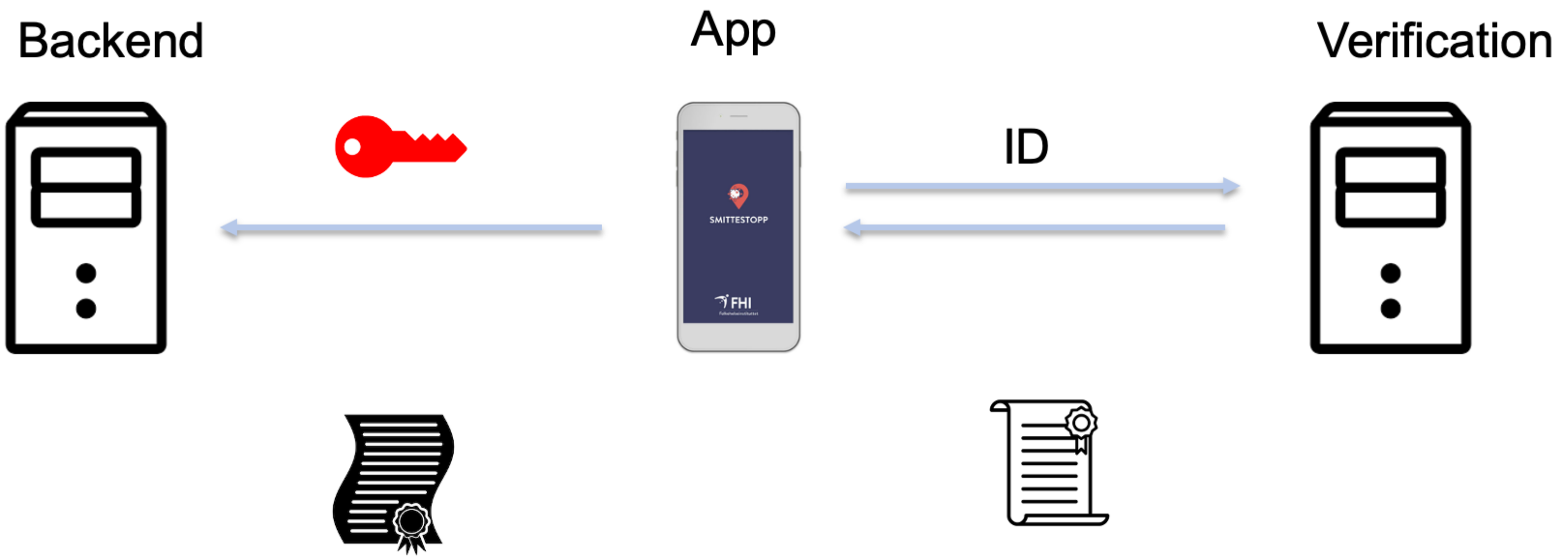


Anonymous Tokens for Private Contact Tracing



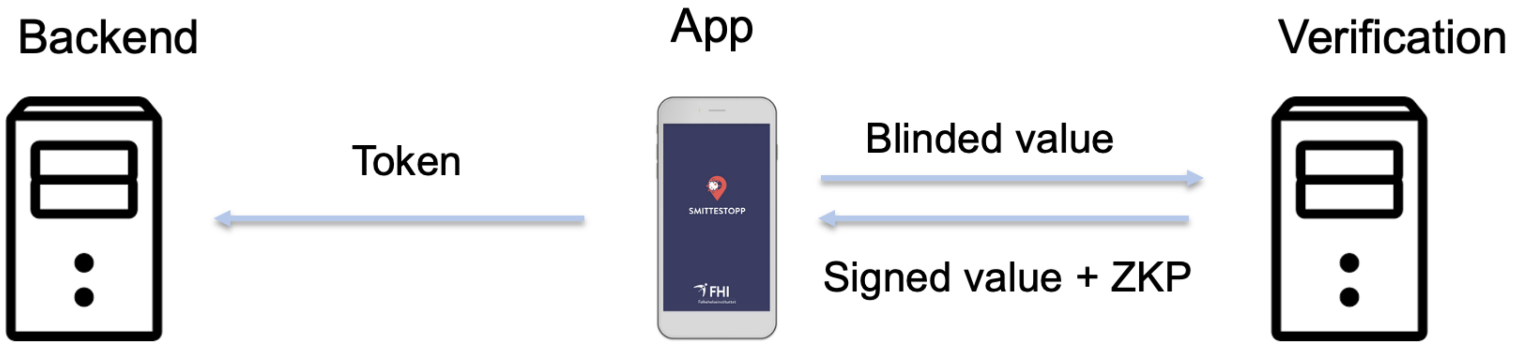
ID can be correlated with the "infection keys"!

Anonymous Tokens for Private Contact Tracing



Solution: The app randomizes the token before it is being forwarded.

Anonymous Tokens for Private Contact Tracing



4. Verify token

1. Choose a random and blinded value to be signed

3. Verify proof and unblind

2. Sign the value, and prove that it was correctly signed

Anonymous Tokens for Private Contact Tracing

Problem: Users should not be able to hold onto a token and upload later. We revoke all unspent tokens older than 3 days.

Solution: The client needs to download new public keys from a public API every time it wants to talk to the server. Impractical.

Note: Still possible to correlate identities with “infection keys” if the servers are logging IP-addresses and timestamps.

Efficiently Revocable Tokens

New anonymous token protocol with public metadata.

Based on ECC, avoids pairings.

Revocation based on metadata.

As efficient as plain Privacy Pass!

	Signing	
User(md, pk)		**Signer**(md, pk, sk)
$d := \mathtt{H}_m(\mathsf{md})$		$d := \mathtt{H}_m(\mathsf{md})$
$U := [d]G + K$		$U := [d+k]G$
$t \leftarrow\!\!\$ \{0,1\}^\lambda, r \leftarrow\!\!\$ \mathbb{Z}_p^*$		$e := (d+k)^{-1}$
$T := \mathtt{H}_t(t)$		
$T' := [r^{-1}]T$	$\xrightarrow{T'}$	$W' := [e]T'$
if not $\mathtt{V}(\pi_{\mathtt{DLEQ}})$	$\xleftarrow{W', \pi_{\mathtt{DLEQ}}}$	$\pi_{\mathtt{DLEQ}} \leftarrow \Pi_{\mathrm{DLEQ}}(G, T', K, W'; e)$
return $\perp$		
$W := [r]W'$		
return (t, md, W)		
	Redemption	
User(t, md, W)		**Verifier**(sk)
	$\xrightarrow{t, \mathsf{md}, W}$	$e := (\mathtt{H}_m(\mathsf{md}) + k)^{-1}$
		if $W = [e]\mathtt{H}_t(t)$
		return true
		else
		return false

Fig. 6. Designated verifier anonymous tokens with public metadata. Our protocol is a direct extension of Privacy Pass [DGS+18].

Efficiently Revocable Tokens

Public Metadata (PM)	PubKey	Request	Signature	Token
Privacy Pass [DGS+18]	$257 \cdot 2^N$	257	769	385
DIT [HIJ+21]	$257 \cdot (N+2)$	257	$769 \cdot (N+1)$	385
Our scheme (Figure 6)	257	257	769	385
PM + Private Metadata	**PubKey**	**Request**	**Signature**	**Token**
Kreuter *et al.* [KLOR20a]	$514 \cdot 2^N$	257	1921	642
Our Scheme (Figure 7)	1028	257	3203	642
PM + Public Verifiability	**PubKey**	**Request**	**Signature**	**Token**
Abe and Fujisaki [AF96]	3202	3072	3072	3200
Our scheme (Figure 8)	763	382	382	510

Table 1. Size given in bits. We compare the schemes for 128 bits of security, allowing for 2^N strings md of metadata. Token seed t is of size 128 bits, and metadata md is implicit knowledge. Privacy Pass, DIT, Kreuter *et al.* and our protocols in Figure 6 and 7 are instantiated with curve x25519 [Ber05], Abe and Fujisaki is instantiated with RSA-3072 and our protocol in Figure 8 is instantiated with BLS12-381 [YCKS21].

Verifiable Shuffles

Goals

1. Build a zero-knowledge protocol to prove correct shuffle of messages
2. Extend the shuffle to handle ciphertexts instead of messages
3. Build a mixing network from the extended shuffle
4. Extend the encryption scheme to support verifiable distributed decryption
5. Combine everything to construct systems for electronic voting
6. Use primitives based on lattices to achieve post-quantum security

Proof of Shuffle

- Public information: sets of commitments $\{[m_i]\}_{i=1}^{\tau}$ and messages $\{\hat{m}_i\}_{i=1}^{\tau}$.
- P knows the openings $\{(m_i, \boldsymbol{r}_{m_i}, f_i)\}_{i=1}^{\tau}$ of the commitments $\{[m_i]\}_{i=1}^{\tau}$,

  and P knows a permutation γ such that $\hat{m}_i = m_{\gamma^{-1}(i)}$ for all $i = 1, ..., \tau$.
- We construct a $4 + 3\tau$-move ZKPoK protocol to prove the statement:

$$R_{\text{Shuffle}} = \left\{ (x, w) \;\middle|\; \begin{array}{l} x = ([m_1], \ldots, [m_\tau], \hat{m}_1, \ldots, \hat{m}_\tau, \hat{m}_i), \\ w = (\gamma, f_1, \ldots, f_\tau, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_\tau), \gamma \in S_\tau, \\ \forall i \in [\tau]: \ \texttt{Open}(\left[m_{\gamma^{-1}(i)}\right], \hat{m}_i, \boldsymbol{r}_i, f_i) = 1 \end{array} \right\}$$

Proof of Shuffle

First, the verifier sends a challenge ρ to shift all commitments and messages $M_i = m_i - \rho$ and $\hat{M}_i = \hat{m}_i - \rho$ to ensure that all messages are invertible.

Secondly, P draws θ_i uniformly at random, and computes the commitments:

$$\begin{aligned} [D_1] &= \left[\theta_1 \hat{M}_1\right] \\ \forall j \in \{2, \ldots, \tau - 1\} : \ [D_j] &= \left[\theta_{j-1} M_j + \theta_j \hat{M}_j\right] \\ [D_\tau] &= [\theta_{\tau-1} M_\tau] \, . \end{aligned} \tag{1}$$

Proof of Shuffle

P receives a challenge β from V and computes s_i such that the following equations are satisfied:

$$\begin{aligned} \beta M_1 + s_1 \hat{M}_1 &= \theta_1 \hat{M}_1 \\ \forall j \in \{2, \ldots, \tau - 1\} : \; s_{j-1} M_j + s_j \hat{M}_j &= \theta_{j-1} M_j + \theta_j \hat{M}_j \\ s_{\tau-1} M_\tau + (-1)^\tau \beta \hat{M}_\tau &= \theta_{\tau-1} M_\tau . \end{aligned} \tag{2}$$

Proof of Shuffle

We can rewrite these equations as a linear system:

$$\begin{bmatrix} M_1 & \hat{M}_1 & 0 & \dots & 0 & 0 \\ 0 & M_2 & \hat{M}_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & M_{\tau-1} & \hat{M}_{\tau-1} \\ (-1)^\tau \hat{M}_\tau & 0 & 0 & \dots & 0 & M_\tau \end{bmatrix} \begin{bmatrix} \beta \\ s_1 \\ \vdots \\ s_{\tau-2} \\ s_{\tau-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

We observe that the determinant of the matrix is equal to $\prod_{i=1}^{\tau} M_i - \prod_{i=1}^{\tau} \hat{M}_j$. If the statement is false, it follows from the Schwartz–Zippel lemma that this system (with high probability) does not have a solution (over the choice of β).

Proof of Shuffle

P uses the protocol Π_{Lin} to prove that each commitment $[D_i]$ satisfies the equations (2). In order to compute the s_i values, we can use the following fact:

Lemma

Choosing

$$s_j = (-1)^j \cdot \beta \prod_{i=1}^{j} \frac{M_i}{\hat{M}_i} + \theta_j \tag{3}$$

for all $j \in 1, \ldots, \tau - 1$ yields a valid assignment for Equation (2).

Proof of Shuffle

Zero-Knowledge Proof Π_{Shuffle} of Correct Shuffle

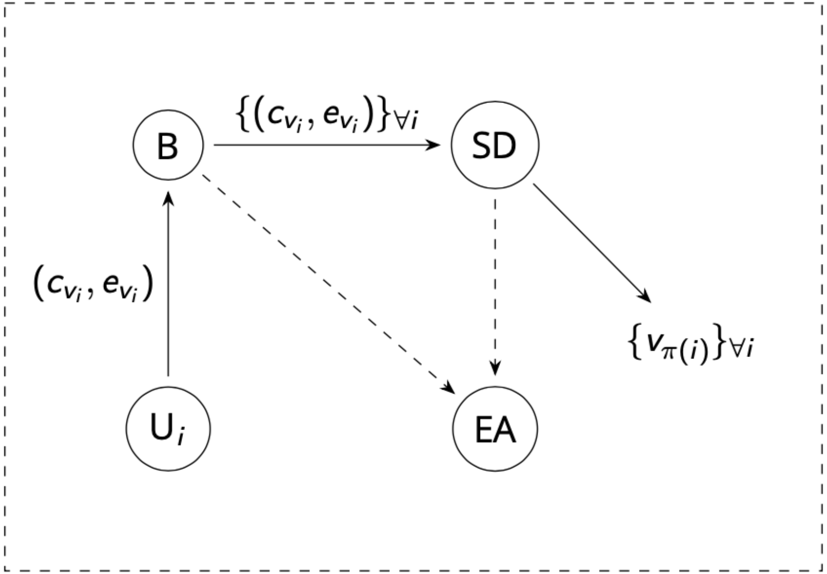
Prover, P		Verifier, V
	$\xleftarrow{\rho}$	$\rho \xleftarrow{\$} R_q \setminus \{\hat{m}_i\}_{i=1}^{\tau}$
$\hat{M}_i = \hat{m}_i - \rho$ $M_i = m_i - \rho$		$\hat{M}_i = \hat{m}_i - \rho$ $[M_i] = [m_i] - \rho$
$\theta_i \xleftarrow{\$} R_q, \forall i \in [\tau - 1]$ Compute $[D_i]$ as in Eq. (1), i.e. $[D_1] = [\theta_1 \hat{M}_1], [D_\tau] = [\theta_{\tau-1} M_\tau],$ $[D_i] = [\theta_{i-1} M_i + \theta_i \hat{M}_i]$ for $i \in [\tau - 1] \setminus \{1\}$	$\xrightarrow{\{[D_i]\}_{i=1}^{\tau}}$	
	$\xleftarrow{\beta}$	$\beta \xleftarrow{\$} R_q$
Compute $s_i, \forall i \in [\tau - 1]$ as in (3).	$\xrightarrow{\{s_i\}_{i=1}^{\tau-1}}$	
		Use Π_{Lin} to prove that (1) $\beta[M_1] + s_1 \hat{M}_1 = [D_1]$ (2) $\forall i \in [\tau - 1] \setminus \{1\} : \; s_{i-1}[M_i] + s_i \hat{M}_i = [D_i]$ (3) $s_{\tau-1}[M_\tau] + (-1)^\tau \beta \hat{M}_\tau = [D_\tau]$ i.e. all equations from (2)

Performance

- Optimal parameters for the commitment scheme is $q \approx 2^{32}$ and $N = 2^{10}$.
- The proof of linearity use Gaussian noise of standard deviation $\sigma_C \approx 2^{15}$.
- The prover sends 1 commitment, 1 ring-element and 1 proof per message.
- The shuffle proof is of total size $\approx 22\tau$ KB for τ messages.
- The shuffle proof takes $\approx 27\tau$ ms to compute for τ messages.

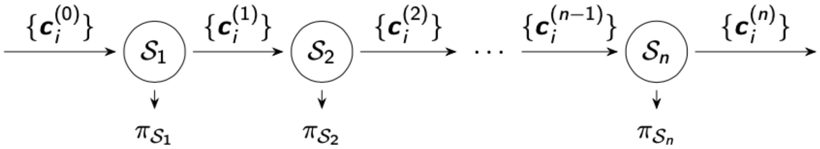
Shuffle-Decryption

- SD both shuffle and decrypt the votes.
- Integrity follows from the ZK-proof.
- Privacy if B and SD does not collude.



Mixing Network

- We extend the shuffle to ciphertexts instead of messages
- We create a mixing network that does the following:
  1. Re-randomize the ciphertexts
  2. Commit to the randomness
  3. Permute the ciphertexts
  4. Prove that shuffle is correct
  5. Prove that the randomness is short
- Integrity follows from the ZK-proofs
- Privacy if at least one server is honest

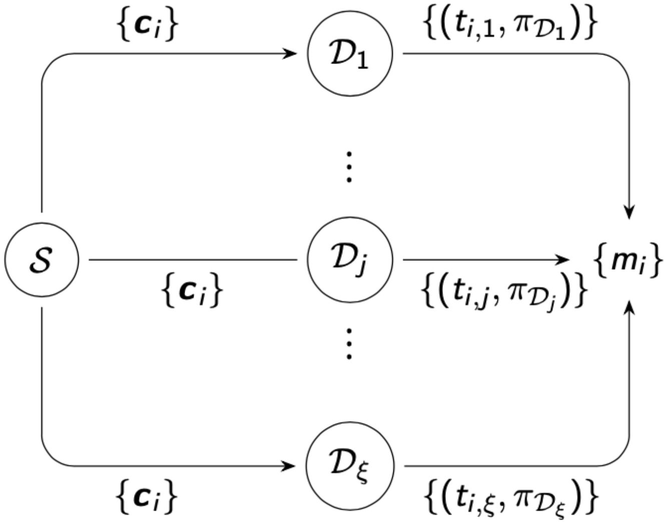


Distributed Decryption

Verifiable distributed decryption protocol:

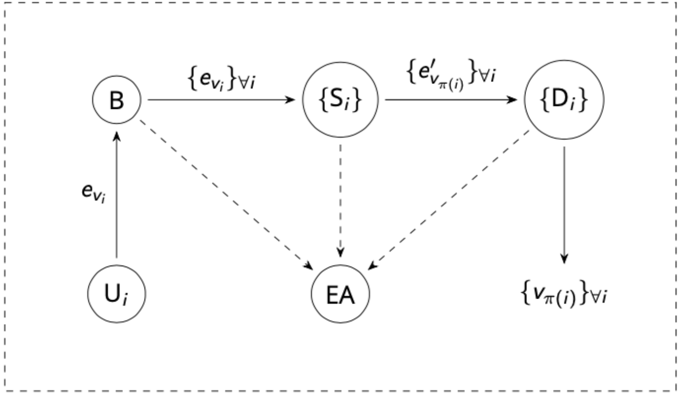
- On input key s_j and ciphertext (u, v), sample large noise E_j, output $t_j = s_j u + pE_j$.
- We use Π_{Lin} to prove correct computation.
- We use Π_{A} to prove that E_j is bounded.

We obtain the plaintext as $m \equiv (v - t \mod q) \mod p$, where $t = t_1 + t_2 + \ldots + t_\xi$.



Mix-Net and Distributed Decryption

- $\{\mathcal{S}_i\}$ may consist of many shuffle-servers.
- $\{\mathcal{D}_i\}$ consists of many decryption-servers.
- Integrity follows from the ZK-proofs.
- Privacy holds if the following is true:
  1. at least one shuffle-server is honest, and
  2. at least one decryption-server is honest.



Performance

Optimal parameters are N = 4096 and q ≈ 2^{78}.

$\boldsymbol{c}_i^{(k)}$	$[\![R_q^\ell]\!]$	π_{Shuf}	$\pi_{L_{i,j}}$	π_{AEx}	π_{ANEx}	$\pi_{\mathcal{S}_i}$	$\pi_{\mathcal{D}_j}$
80 KB	$40(\ell+1)$ KB	150τ KB	35 KB	20τ KB	2τ KB	370τ KB	157τ KB

Table 3. Size of the ciphertexts, commitments and proofs.

Performance

Protocol	$\Pi_{\text{LIN}} + \Pi_{\text{LINV}}$	$\Pi^{\ell}_{\text{SHUF}} + \Pi^{\ell}_{\text{SHUFV}}$	$\Pi_{\text{ANEx}} + \Pi_{\text{ANExV}}$	$\Pi_{\text{AEx}} + \Pi_{\text{AExV}}$
Time	$(10.7 + 15.7)\tau$ ms	$(15.1 + 16.1)\tau$ ms	$(30.0 + 25.0)\tau$ ms	$(1009 + 20)\tau$ ms

Table 5. Timings for cryptographic protocols, obtained by computing the average of 100 consecutive executions with $\tau = 1000$.

Verifiable Decryption

Lattice-Based Verifiable Decryption

A verifiable decryption protocol is a zero-knowledge protocol proving that a certain message is the correct decryption of a certain ciphertext with respect to a committed key which does not reveal anything about the decryption key.

Verifiable decryption is crucial to prove correct outcome in electronic voting. Today's systems use discrete logs, and can be broken by quantum computers.

Goal: design an efficient verifiable decryption protocol for lattice cryptography.

Verifiable Decryption in the Head

1. Deal splits the key into two parts and prove correctness.
2. Play compute a decryption share $t_{i,j}$ based on key share s_i.
3. P commits to the shares, and V challenges half of them.
4. V verifies all shares.
5. V reconstructs to check the message from the shares.

Π_{ZKPCD}

Prover$((\mathsf{pk}, \{c_j\}_{j=1}^{\tau}, \{m_j\}_{j=1}^{\tau}), (\mathsf{sk}))$

$k = 1, \ldots, \lambda:$

$\quad (\mathsf{sk}_{0,k}, \mathsf{sk}_{1,k}, \mathsf{aux}_k) \leftarrow \mathsf{Deal}(\mathsf{pk}, \mathsf{sk})$

$\quad i = 0, 1, j = 1, \ldots, \tau:$

$\qquad t_{i,j,k} \leftarrow \mathsf{Play}(\mathsf{sk}_{i,k}, c_j; \rho_{i,k,j})$

$w \leftarrow (\{\mathsf{aux}_k, \{t_{i,j,k}\}\})$

$\xrightarrow{w}$

Verifier$(\mathsf{pk}, \{c_j\}_{j=1}^{\tau}, \{m_j\}_{j=1}^{\tau})$

$\beta \xleftarrow{\$} \{0,1\}^{\lambda}$

$\xleftarrow{\beta}$

$z \leftarrow (\{\mathsf{sk}_{\beta[k],k}\}_k, \{\rho_{\beta[k],k,j}\}_{k,j})$

$\xrightarrow{z}$

$k = 1, \ldots, \lambda:$

$\quad \mathsf{Verify}(\mathsf{pk}, \mathsf{aux}_k, \beta[k], \mathsf{sk}_{\beta[k],k}) \stackrel{?}{=} 1$

$\quad j = 1, \ldots, \tau:$

$\qquad \mathsf{Play}(\mathsf{sk}_{\beta[k],k}, c_j; \rho_{\beta[k],k,j}) \stackrel{?}{=} t_{\beta[k],j,k}$

$\qquad \mathsf{Reconstruct}(c_j, t_{0,j,k}, t_{1,j,k}) \stackrel{?}{=} m_j$

Verifiable Decryption in the Head

Parameter	Explanation	Constraints	Value
N	Dimension	Power of two	2048
q	Ciphertext modulus	$B_{\mathtt{Dec}} \ll q \equiv 1 \mod 2N$	$\approx 2^{55}$
p	Plaintext modulus		2
κ	Security parameter	Long-term privacy	128
$\mathtt{sec}$	Statistical security		40
λ	Soundness parameter		10, ..., 128
μ	Repetitions of Π_{ZKPoS}	$\mu \geq \lambda \cdot \ln(2)/\ln(3/2)$	17, ..., 218
B_∞	Bounds on secrets		1
$B_{\mathtt{Dec}}$	Decryption bound	$\|v - su\|_\infty \leq B_{\mathtt{Dec}}$	$\approx 2^{13}$
Size of π_D	Timings for π_D	Size of π_S	Timings for π_S
$14\lambda\tau$ KB	$4\lambda\tau$ ms	$175\lambda\mu$ KB	$30\lambda\mu$ ms

Table 1. Notation, explanation, constraints and concrete parameters for the protocol. We also provide size and timings for decryption proof π_D and proofs of shortness π_S.

Verifiable Decryption for BGV

The verifiable decryption protocol Π_{Dec}, for prover $\mathcal{P}$, goes as following:

1. $\mathcal{P}$ takes as input a set of ciphertexts $(u_1, v_1), \ldots, (u_\tau, v_\tau)$ and $([\![s]\!], s, \boldsymbol{r}_s, f_s)$.
2. $\mathcal{P}$ runs $\mathtt{Dec}$ on input s and (u_i, v_i) for all $i \in [\tau]$ to obtain $m_1, \ldots, m_\tau$.
3. $\mathcal{P}$ extracts noise d_i by computing $d_i = (v_i - m_i - u_i s)/p \mod q$ for all $i \in [\tau]$.
4. $\mathcal{P}$ commits to all d_i as $[\![d_i]\!]$, and proves $p[\![d_i]\!] = v_i - m_i - u_i[\![s]\!]$ using Π_{Lin}.
5. $\mathcal{P}$ uses protocol Π_{A} to prove that all $\|d_i\|_2$ are bounded by $B_{\mathsf{A}} \leq \sqrt{2vN}\sigma_{\mathsf{A}}$.
6. $\mathcal{P}$ outputs messages $\{m_i\}_{i=1}^{\tau}$, commitments $\{[\![d_i]\!]\}_{i=1}^{\tau}$, proofs $\{\pi_{L_i}\}_{i=1}^{\tau}, \pi_{\mathsf{A}}$.

Verifiable Decryption for BGV

Message m_i	Ciphertext (u_i, v_i)	Commitment $[\![d_i]\!]$	Proof π_{L_i}	Proof π_{A}	Proof π_{DEC}
0.256 KB	22.6 KB	22.6 KB	19 KB	2τ KB	43.6τ KB

Table 2. Sizes for parameters $p = 2, q \approx 2^{44}$ and $N = 2048$ computing proof $\pi_{\mathrm{DEC}} = (\{[\![d_i]\!], \pi_{L_i}\}_{i=1}^{\tau}, \pi_{\mathrm{A}})$, where shortness proofs π_{A} is amortized over batches of size 2048.

Noise $[\![d_i]\!]$	Proof Π_{LIN}	Verification Π_{LINV}	Proof Π_{A}	Verification Π_{AV}	Proof π_{DEC}
5τ ms	47τ ms	12τ ms	24τ ms	12τ ms	76τ ms

Table 3. Amortized time per instance over $\tau = 2048$ ciphertexts.

Summary & Conclusions

Privacy matters: it is a human right; it is protected by law (GDPR); it allows people to be themselves. We need to build systems that protects privacy.

Quantum computers are being built as we speak, and NIST is standardizing quantum secure key encapsulations mechanisms and digital signatures. We need to build an ecosystem of quantum secure crypto for real-world use.

THANK YOU!