GDAŃSK UNIVERSITY
OF TECHNOLOGY

Compromise-Resistant Protocols for Sensitive Services in Distributed Systems

Stanisław Barański

2026

Context: short story about control and data

- Everyone needs more information: governments to protect citizens (health, safety, justice), companies to improve services, and individuals to make better decisions.

Context: short story about control and data

- Everyone needs more information: governments to protect citizens (health, safety, justice), companies to improve services, and individuals to make better decisions.
- Information is power; centralized data often leads to centralized power.

Context: short story about control and data

- Everyone needs more information: governments to protect citizens (health, safety, justice), companies to improve services, and individuals to make better decisions.
- Information is power; centralized data often leads to centralized power.
- The question is **not "should we** use more data?", **but "how** can we use it responsibly" with strong, verifiable guarantees.

Context: short story about control and data

- Everyone needs more information: governments to protect citizens (health, safety, justice), companies to improve services, and individuals to make better decisions.
- Information is power; centralized data often leads to centralized power.
- The question is **not "should we** use more data?", **but "how** can we use it responsibly" with strong, verifiable guarantees.
- Today, these guarantees are enforced mainly by regulation and institutional trust, but that trust can fail (breaches, abuse, regime change).

Context: short story about control and data

- Everyone needs more information: governments to protect citizens (health, safety, justice), companies to improve services, and individuals to make better decisions.
- Information is power; centralized data often leads to centralized power.
- The question is **not "should we** use more data?", **but "how** can we use it responsibly" with strong, verifiable guarantees.
- Today, these guarantees are enforced mainly by regulation and institutional trust, but that trust can fail (breaches, abuse, regime change).
- The AI wave raises the stakes: systems trained on sensitive data should be auditable, fair, and privacy-preserving by design.

Context: short story about control and data

- Everyone needs more information: governments to protect citizens (health, safety, justice), companies to improve services, and individuals to make better decisions.
- Information is power; centralized data often leads to centralized power.
- The question is **not "should we** use more data?", **but "how** can we use it responsibly" with strong, verifiable guarantees.
- Today, these guarantees are enforced mainly by regulation and institutional trust, but that trust can fail (breaches, abuse, regime change).
- The AI wave raises the stakes: systems trained on sensitive data should be auditable, fair, and privacy-preserving by design.
- Privacy is not only a preference: it supports progress and social stability by reducing surveillance and coercion, while enabling safe data use (e.g., healthcare, AI).

This dissertation addresses how to build trustworthy and privacy-preserving systems for sensitive services using distributed trust and cryptographic guarantees.

Trust math and mutually distrusting parties, not a single authority

- Trust in TTPs is fragile: systems get compromised, and policies can change over time.

Trust math and mutually distrusting parties, not a single authority

- Trust in TTPs is fragile: systems get compromised, and policies can change over time.
- Cryptography can enforce privacy and verifiability independently of institutional trust.
  - In particular: VPPC is the family of mechanisms that let parties compute on sensitive inputs while still being able to convince others the computation was correct without revealing the inputs (ZK proofs, MPC/threshold crypto, HE, TEE+attestation, DP).

Trust math and mutually distrusting parties, not a single authority

- Trust in TTPs is fragile: systems get compromised, and policies can change over time.
- Cryptography can enforce privacy and verifiability independently of institutional trust.
  - In particular: VPPC is the family of mechanisms that let parties compute on sensitive inputs while still being able to convince others the computation was correct without revealing the inputs (ZK proofs, MPC/threshold crypto, HE, TEE+attestation, DP).
- Distributed Ledger Technologies (DLT) enable public coordination and audit without a single operator.
  - DLT is a replicated, append-only public log/state maintained by distributed consensus. It gives us a shared public transcript (T_{pub}): a globally visible sequence of protocol events (e.g., commitments, ciphertexts, proofs, signatures, receipts, and ledger state updates) that all parties can read and verify.

Trust math and mutually distrusting parties, not a single authority

- Trust in TTPs is fragile: systems get compromised, and policies can change over time.
- Cryptography can enforce privacy and verifiability independently of institutional trust.
  - In particular: VPPC is the family of mechanisms that let parties compute on sensitive inputs while still being able to convince others the computation was correct without revealing the inputs (ZK proofs, MPC/threshold crypto, HE, TEE+attestation, DP).
- Distributed Ledger Technologies (DLT) enable public coordination and audit without a single operator.
  - DLT is a replicated, append-only public log/state maintained by distributed consensus. It gives us a shared public transcript (T_{pub}): a globally visible sequence of protocol events (e.g., commitments, ciphertexts, proofs, signatures, receipts, and ledger state updates) that all parties can read and verify.
- Combined: abuse becomes computationally infeasible, or requires compromise of many independent parties (often economically infeasible).

Sensitive services

- **Sensitive services** process protected data/relations (legal, regulatory, business, or personal sensitivity).
- In this dissertation we focus on the following set of five sensitive service classes:

$$\mathcal{S}_{sens} = \{s_{\text{ivot}}, s_{\text{ppml}}, s_{\text{iot}}, s_{\text{anonser}}, s_{\text{mixer}}\}.$$

- Each service instance s is identified by:
  (F_s, T_{pub}, protected secrets/links, $\mathcal{R}_s$, threat model).
  - F_s: the ideal functionality (e.g., tally votes, aggregate data, mix funds).
  - T_{pub}: the public transcript (commitments, proofs, receipts).
  - Protected secrets/links: private inputs or relationships that must not be revealed.
  - $\mathcal{R}_s$: the set of roles/parties involved (voters, servers, clients, etc.).
  - Threat model: assumptions about adversarial capabilities and compromise thresholds.
- We model a service as a stateful ideal functionality:

$$(y_1, \ldots, y_n, T'_{\text{pub}}) \leftarrow F_s(x_1, \ldots, x_n, T_{\text{pub}}).$$

Security properties

- Each sensitive service requires a specific set of security properties. We focus on three core properties:
  - **Correctness (CORR):** outputs correspond to the intended logic and public transcript.
  - **Privacy (PRIV):** protected inputs/links are not revealed beyond what is implied by public transcript and legitimate outputs.
  - **Liveness (LIVE):** requests eventually produce an output or a terminal abort.

What do I mean by compromise?

- In this dissertation, **compromise** is an umbrella term.
- It means that a trust domain can no longer be treated as independently enforcing the protocol's intended guarantees.
- This includes:
  - adversarial corruption,
  - Byzantine deviation,
  - insider abuse or intentional misbehavior,
  - bribery or coercion,
  - collusion with other domains,
  - leakage or theft of keys/secrets,
  - unintentional loss of control over infrastructure, if it gives equivalent effective power.
- So the question is not only "who can hack the system?", but more generally:
- **what is the smallest set of trust domains whose effective loss of independence or control can break CORR, PRIV, or LIVE?**

Role of VPPC and DLT

- VPPC enforces that outputs (y_i) correspond to the *committed private inputs* under F_s, without revealing inputs (x_i) or protected links.
- DLT enforces a single, consensus-ordered and tamper-evident public transcript T_{pub} of submitted inputs and outputs/proofs, so everyone agrees what was provided and can audit that only valid, verified state transitions were accepted.
- The main challenge is that:
  - DLT can enforce **Correctness** and **Liveness** for what is **publicly** visible and checkable on T_{pub},
  - but sensitive services need **Privacy** and **Correctness** for **private** inputs, hence the need for VPPC.
- We focus on adversarial (Byzantine) environments: compromised parties may arbitrarily deviate from the protocol; crash/abort-only failures are substantially easier.
- The hard part is achieving **PRIV** and **CORR** simultaneously: each is manageable in isolation, but together they require proving correct processing of private inputs without revealing them, even under Byzantine behavior.

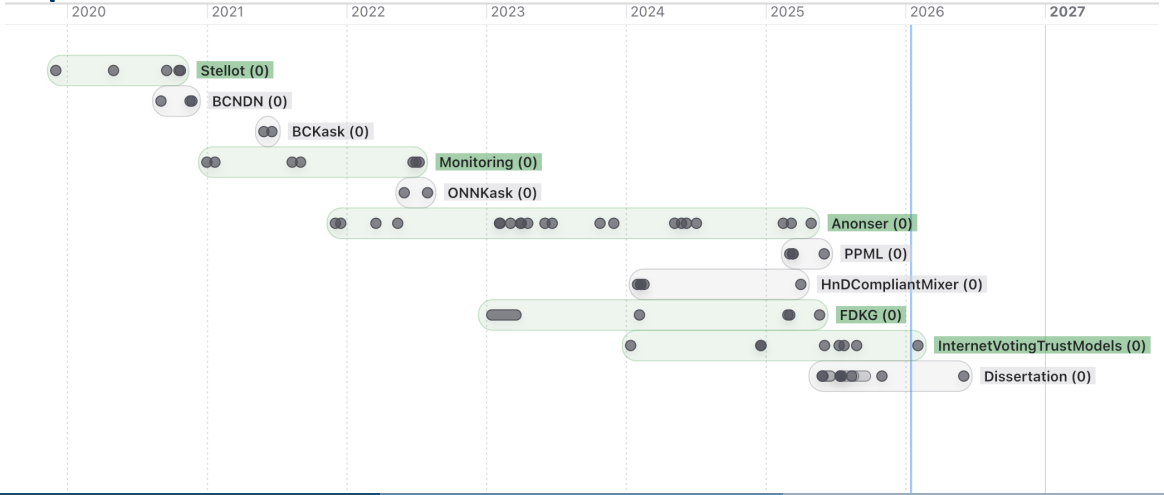
Protocols for sensitive services using VPPC and DLT

- $\mathcal{P}_{\mathrm{VPPC+DLT}}$ is the set of protocols to realise sensitive services that combine:
  - a DLT layer to coordinate and provide the immutable public transcript (T_{pub}), and
  - a VPPC layer to do sensitive computation off-chain (or privately) and output proofs/attestations that the DLT can verify.

Protocols for sensitive services using VPPC and DLT

- $\mathcal{P}_{\mathrm{VPPC+DLT}}$ is the set of protocols to realise sensitive services that combine:
  - a DLT layer to coordinate and provide the immutable public transcript (T_{pub}), and
  - a VPPC layer to do sensitive computation off-chain (or privately) and output proofs/attestations that the DLT can verify.
- We demonstrate a diverse set of protocols $\mathcal{P}_{\mathrm{VPPC+DLT}}$:
  - Internet voting protocol (Applied Sciences, 2020, 100pkt) [1].
  - Privacy-Preserving, Scalable Blockchain-Based Monitoring (Appl Sci, 2022, 100pkt) [2].
  - Compliant Privacy Mixers (IACR preprint, 2025, 0pkt) [3].
  - Anonymous Provision of Privacy-Sensitive Services (IJIS, 2025, 100pkt) [4].
  - FDKG-based internet voting protocol (FGCS, 2025, 140pkt) [5].
  - Privacy Preserving Machine Learning (TASKQ, 2025, 20pkt) [6], [7].

First 5 years of my PhD: Sensitive services, Protocols, Implementations, and Articles



Sensitive services and protocol variants

Service $s \in \mathcal{S}_{sens}$		Protocol variants π (**witnesses for Thesis 1**)	Source
s_{ivot}	Internet Voting	p_1 centralized distributor + centralized key-holder	[pub]
		p_2 **threshold distributor + threshold key-holder**	[diss]
		p_3 **FDKG-based voting**	[pub]
s_{ppml}	PP Machine Learning	p_1 HE-BC	[diss]
		p_2 TEE-BC	[diss]
		p_3 **PPML-BC***	[diss]
s_{iot}	IoT Monitoring	p_1 single key-holder + single aggregator	[pub]
		p_2 **many key-holders + MPC aggregator**	[diss]
s_{mixer}	Compliant Mixer	p_1 Haze	[pub]
		p_3 **Haze-thresholded**	[diss]
s_{anonser}	Anonymous Services	p_1 baseline + single court	[pub]
		p_2 baseline + DRS (e.g., Kleros)	[pub]
		p_3 **Anonser-BC**^{MV} **(digital-verifiable + multi-provider)**	[diss]

[pub] published / implemented earlier [diss] variant defined in this dissertation

Thesis 1

Combining VPPC with DLT enables building protocols for sensitive services that guarantee **CORR**, **PRIV**, and **LIVE**, without a single trusted third party.

Thesis 1

- Combining VPPC with DLT enables building protocols for sensitive services that guarantee **CORR**, **PRIV**, and **LIVE**, without a single trusted third party.
- Formal statement:

$$\forall s \in \mathcal{S}_{sens}\ \exists \pi \in \mathcal{P}_{\mathrm{VPPC+DLT}}\ \forall \phi \in \{\mathrm{CORR}, \mathrm{PRIV}, \mathrm{LIVE}\} :\ \mathrm{CRE}_{\phi}(\pi, s) > 1.$$

- For each $s \in \mathcal{S}_{sens}$ we provide at least one **witness protocol** π (bold variants) and a CRE evaluation showing $\mathrm{CRE}_{\phi}(\pi, s) > 1$ for all three properties.

To support Thesis 1 we

- Identify five distinct classes of sensitive services—internet voting, privacy-preserving machine learning, anonymous service provision, infrastructure monitoring, and compliant privacy mixers—and formalize their goals, threat models, and security requirements.
- Design a concrete BC+VPPC protocol for each service class, showing how blockchain and verifiable privacy-preserving computation can be combined to realize the required functionality without relying on a single trusted operator.
- Develop CRE, a unified and reproducible methodology for analyzing how much trust-domain compromise is needed to violate correctness, privacy, or liveness in a distributed protocol.
- Apply CRE to the designed witness protocols and show that, for each service class, there exists at least one protocol variant satisfying the claimed predicate $\mathrm{CRE}_\phi > L^{(1)}$ for all three core properties $\phi \in \{\mathrm{CORR}, \mathrm{PRIV}, \mathrm{LIVE}\}$.

Compromise Resistance Evaluation (CRE)

- CRE maps protocols to a **compromise-resistance profile**.
- It measures how much trust-domain compromise is needed to violate selected security properties.
- In simple words: **what is the minimal compromise needed to break each property?**
- CRE is inspired by the earlier TMQ framing, but uses attacker-centered language that better matches the actual analysis.
- In this work,

$$\mathrm{CRE} : \mathcal{P}_{\mathrm{VPPC+DLT}} \to \{L_{\mathrm{CORR}}, L_{\mathrm{PRIV}}, L_{\mathrm{LIVE}}\}.$$

$$\mathrm{CRE} : \mathcal{P}_{\textbf{VPPC+DLT}} \to \{L_{\mathrm{CORR}}, L_{\mathrm{PRIV}}, L_{\mathrm{LIVE}}\}$$

- **Correctness**: a compromise set can make an honest party accept an output inconsistent with the intended logic and public transcript (without cryptographic breaks).
- **Privacy**: a compromise set can learn protected inputs (or a protected link) beyond what is implied by public transcript and legitimate outputs.
- **Liveness**: a compromise set can keep an honest request pending indefinitely (no output and no terminal abort), by withholding an action on the critical path.
- L_ϕ is the level induced by the weakest minimal compromise set that can break property ϕ.

L_ϕ — level of the weakest minimal compromise set that can break property ϕ

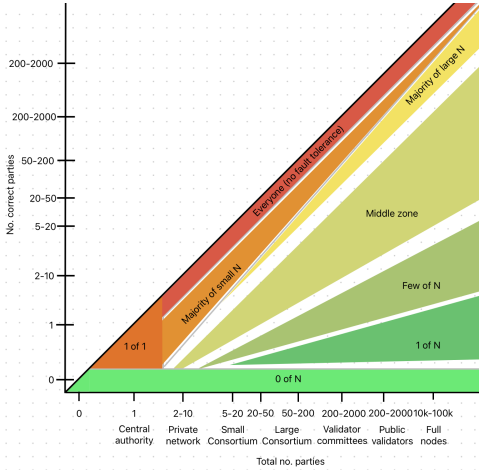
- $L_\phi = k$ means the weakest successful compromise pattern for property ϕ falls into category k.

L_ϕ — level of the weakest minimal compromise set that can break property ϕ

- $L_\phi = k$ means the weakest successful compromise pattern for property ϕ falls into category k. **In protocol analysis we often do not know exact cardinalities in a literal sense, so we use structural categories.**

L_ϕ — level of the weakest minimal compromise set that can break property ϕ

- $L_\phi = k$ means the weakest successful compromise pattern for property ϕ falls into category k. **In protocol analysis we often do not know exact cardinalities in a literal sense, so we use structural categories.**
- $L_\phi = 1$ — one trust domain suffices (centralized trust).
- $L_\phi = 2$ — a small fixed set of named roles must all be compromised.
- $L_\phi = 3$ — a threshold of a fixed committee must be compromised.
- $L_\phi = 4$ — an open consensus majority (e.g., validators) must be compromised.
- $L_\phi = 5$ — no trust-domain compromise suffices; only a cryptographic failure.



CRE Evaluation Procedure I

For each protocol π we apply the following procedure to obtain

$$\mathrm{CRE}(\pi) = \{L_{\mathrm{CORR}}(\pi), L_{\mathrm{PRIV}}(\pi), L_{\mathrm{LIVE}}(\pi)\}.$$

The steps are written to support independent replication and inter-rater reliability.

1. **Step 0: Fix background assumptions.**
2. **Step 1: Collect source material and resolve ambiguities.**
3. **Step 2: Identify the threat model and define security properties (CORR/PRIV/LIVE).**
4. **Step 3: Identify trust domains $\mathcal{R}_A$.**
5. **Step 4: Specify the ideal functionality and the public transcript interface. Write $(y_1, \ldots, y_n, T'_{\mathrm{pub}}) \leftarrow F_s(x_1, \ldots, x_n, T_{\mathrm{pub}})$.**

CRE Evaluation Procedure II

6. **Step 5: For each trust domain $i \in \mathcal{R}_A$, write** $\text{View}_i = (x_i, r_i, m_i^{\text{in/out}}, T_{\text{pub}})$.
7. **Step 6: Define the goal formulas $G_X(A)$ for $X \in \{\text{CORR}, \text{PRIV}, \text{LIVE}\}$.**
   Write, for each property, a short operational scope paragraph, then give a Boolean goal formula over primitives. Goals must be stated in terms of *Read/Write/Withhold* only.
8. **Step 7: Compute minimal compromise sets by satisfying goals.**
   For each property X:
   - enumerate candidate compromise sets S and prove $\left(\bigcup_{i \in S} \text{Cap}(i)\right) \models G_X(A)$;
   - inclusion-minimise these sets to obtain $\mathcal{M}_X(A)$.
9. **Step 8: Classify each minimal set into a level and take the weakest.**
   For each $S \in \mathcal{M}_X(A)$ map its structure to $\ell(S) \in \mathcal{L}$. Then $L_X(A) = \min_{S \in \mathcal{M}_X(A)} \ell(S)$.
10. **Step 9: Evaluate centralized counterpart (when applicable).**
11. **Step 10: Multi-rater application (IRR).**

What we show for Thesis 1

- **Assumed (VPPC building blocks):**
  - VPPC mechanisms used by a variant satisfy their standard security definitions (e.g., soundness/zero-knowledge, privacy/correctness of MPC/threshold crypto, correctness of HE, validity of TEE attestation).
- **Shown in this dissertation (CRE):**
  - For each protocol variant π we define trust domains, views, and goal formulas for CORR/PRIV/LIVE.
  - We derive **minimal compromise sets** that can break each property, and map them to levels $L_{\mathrm{CORR}}(\pi), L_{\mathrm{PRIV}}(\pi), L_{\mathrm{LIVE}}(\pi)$.
- **For variants marked [diss] in the table:** we additionally provide a precise protocol description and explicit assumptions, then apply the same CRE procedure at the capability level.

Results for Thesis 1

$$\forall s \in \mathcal{S}_{sens}\ \exists \pi \in \mathcal{P}_{\textbf{VPPC+DLT}}\ \forall \phi \in \{\text{corr}, \text{priv}, \text{live}\} :\ \text{TMQ}_{\phi}(\pi, s) > 1$$

Service / Variant	CORR	PRIV	ROB	PATTERNS
Centralized Stellot	1	1	1	BLINDSIG, NULL, PKE, MERKLE, REG
Stellot^{+}	4	1	1	POE, BLINDSIG, NULL, PKE, MERKLE, REG
Stellot*	4	3	1	POE, BLINDSIG, NULL, PKE, MERKLE, REG,
Stellot†	4	3	3	POE, BLINDSIG, NULL, PKE, MERKLE, REG,
FDKG-Vote-BC	4	3	3	CPV, TC, LHE, RANGE
Centralized PPML-BC-VPPC	1	1	1	–
PPML-TEE-BC-TEE	1	2	1	POE, CPV, PKE
PPML-HE-BC-HE	1	5	1	POE, PKE
PPML-BC*-BC-HE-zkML	4	5	2	POE, CPV, PKE
Centralized mixer	1	1	1	–
CompliantMixer-BC$_{\text{cen}}$ Daze	4	1	1	CPV, MERKLE, NULL, REG, PKE
CompliantMixer-BC$_{\text{cen}}$ Haze	4	5	1	CPV, MERKLE, NULL, REG
CompliantMixer-BC$_{\text{thr}}$ Daze	4	3	3	CPV, MERKLE, NULL, REG, PKE, TC
CompliantMixer-BC$_{\text{thr}}$ Haze	4	5	3	CPV, MERKLE, NULL, REG, TC
Centralized Anonser	1	5	1	PKE, DISPUTE, EVID
Anonser-BC	1	5	1	POE, PKE, DISPUTE, EVID
Anonser-BC^{M}	3	5	2	POE, PKE, DISPUTE, EVID
Anonser-BC$^{\text{VPPC}}$	4	5	1	POE, CPV, PKE, DISPUTE, EVID
Anonser-BC^{MV}	4	5	2	POE, CPV, PKE, DISPUTE, EVID

Thesis 2: Avoiding central bottlenecks with threshold cryptography

- Many $\mathcal{P}_{\mathrm{VPPC+DLT}}$ need a key operation on the critical path: *decrypt tally*, *sign authorization*, *release secret*, *rotate committee key*.
- Replacing a single key holder with Distributed Key Generation (DKG) moves the weakest successful compromise pattern from *one party* to a *threshold of parties*.

$$L_{\mathrm{PRIV}}(\pi_{\mathrm{DKG}}, s) \;=\; 3 \quad \text{whereas} \quad L_{\mathrm{PRIV}}(\pi_{\mathrm{single}}, s) = 1.$$

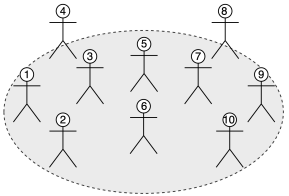
- However, classical (t, n)-DKG is not scalable in open / large networks because it assumes a fixed set and high timely participation; when participation drops, liveness degrades or setup aborts.
- "Getting 60% of 10 people online is easy; getting 60% of 1,000 people online in the same window is hard."

Thesis 2: Problem of DKG in large/open networks

- **Distribution/setup phase (classical DKG):** n and t are fixed before execution; setup assumes the declared participant set is present. If realized setup participation is $n' < n$, setup aborts and must restart.
- **Reconstruction/use phase (classical DKG):** threshold operations need at least t available shares; if fewer than t parties are reachable, reconstruction is blocked.
- The bottleneck is phase-specific: **mandatory participation** in setup and a **global threshold quorum** in reconstruction.

Thesis 2

- Published in Federated Distributed Key Generation (FGCS, 2025, 140pkt) [5].
- FDKG replaces one global quorum with many local quorums (guardian sets), enabling large-n deployments with optional participation, while keeping one-round PVSS-style broadcasts.
- **Distribution (setup) improvement:** FDKG allows optional participation: any realized dealer set D (up to eligible n) can complete setup without requiring all declared participants online.
- **Reconstruction improvement:** each dealer contribution is recoverable from its guardian set, so recovery depends on local availability rather than one global committee quorum.

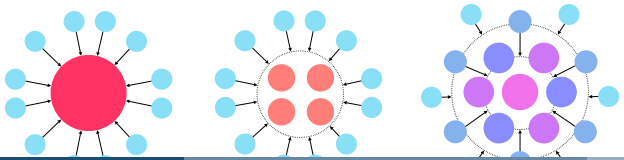


FDKG: distribution and reconstruction

- **Setup/distribution:** classical DKG fixes one declared participant set and expects all of them during setup; if realized participation is lower, setup aborts and retries.
- FDKG setup uses a realized set $D \subseteq \{1, \ldots, n\}$, so absent non-participants do not force a global restart.
- **Reconstruction:** each dealer $i \in D$ chooses guardians G_i (size k) and local threshold t.
- Recovery succeeds if every contribution is locally recoverable:

$$\forall i \in D: \ \big(i \text{ reveals}\big) \ \vee \ \big(|T \cap G_i| \geq t\big).$$

- With reliability-aware guardian selection, recoverable sets are smaller and churn tolerance is higher; coordination scales with k (local), not n (global).



What we show for Thesis 2

- **DKG:** Replacing a single key holder with DKG-based threshold cryptography increases L_{PRIV} and often L_{LIVE}.
- **Distribution phase:** We show FDKG setup does not require all declared participants online; it proceeds with the realized dealer set D.
- **Reconstruction phase:** We show local guardian recoverability tolerates higher churn than one global threshold, especially when guardians are more reliable nodes.
- **Security:** We provide formal proofs FDKG is secure (correctness, privacy, and liveness) under standard assumptions.
- **Scalability:** Simulations show higher success probability than classical DKG under probabilistic participation/retention.
- **Cost:** FDKG keeps one-round broadcast in the setup phase; communication scales as $O(|D| \cdot k)$.
- **Application:** We integrate FDKG with internet voting protocol.

Results for Thesis 2

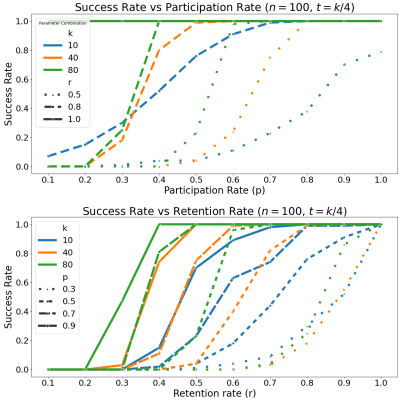


Figure: Success vs. participation p (top) and retention r (bottom) at $n = 100$ with $t = k/4$. Line styles encode the

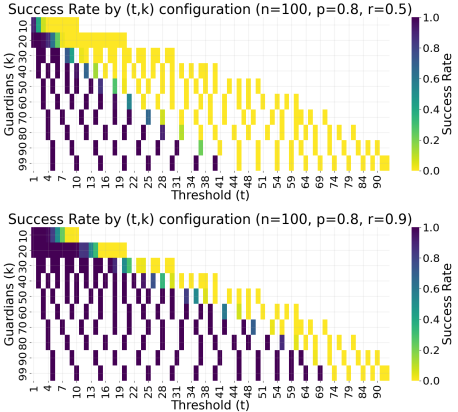


Figure: Success over (k, t) for $n = 100$, $p = 0.8$, with $r = 0.5$ (top) and $r = 0.9$

Main contributions

- **A set of sensitive services and protocol variants.**
  - Defined $\mathcal{S}_{sens} = \{s_{\text{ivot}}, s_{\text{ppml}}, s_{\text{iot}}, s_{\text{anonser}}, s_{\text{mixer}}\}$ and, for each s, designed a set of $\mathrm{DLT{+}VPPC}$ protocol variants with explicit roles, transcripts, and assumptions.
  - Identified **witness variants** (bold in the table) that achieve $\mathrm{CRE}_{\phi}(\pi, s) > 1$ for $\phi \in \{\mathrm{CORR}, \mathrm{PRIV}, \mathrm{LIVE}\}$.
- **Compromise Resistance Evaluation (CRE): a replicable method to compare architectures.**
  - Introduced a step-by-step evaluation pipeline: Design $\rightarrow$ Views $\rightarrow$ Capabilities $\rightarrow$ Goal formulas $\rightarrow$ Minimal compromise sets $\rightarrow$ Levels.
  - Produced a **CRE coding sheet** for each variant and a summary profile $\mathrm{CRE}(\pi) = \{L_{\mathrm{CORR}}, L_{\mathrm{PRIV}}, L_{\mathrm{LIVE}}\}$ enabling cross-protocol comparison.
- **New cryptographic primitive for open/large networks: Federated DKG (FDKG).**
  - Designed and analyzed FDKG: replaces one global quorum with many local guardian quorums, enabling optional participation and heterogeneous trust.
  - Demonstrated that replacing a single key holder with (F)DKG strengthens **PRIV** and often **LIVE** by raising the compromise-resistance level.
- **Practical evidence: protocols and implementations.**

Limitations

- We do not provide a single unified architecture that achieves all properties for all sensitive services.
- The security properties are evaluated at a high (capability) level.
- Some level boundaries (1..5) can be blurry in edge cases (e.g., censoring vs rewriting, "few/1" vs "closed t/n").
- For the thesis, I only need the claim $\mathbf{CRE}_{\phi} > 1$, so ultra-sharp borders beyond that are not essential.

Status & next steps (roadmap to finish the dissertation)

- **Step 1: Freeze scope and assumptions.**
  - lock $\mathcal{S}_{sens}$, the evaluated protocol variants, and the operational scope of CORR/PRIV/LIVE;
  - lock background assumptions (network model, "no cryptographic breaks", meaning of terminal abort).
- **Step 2: Finish CRE coding for each protocol variant.**
  - for each variant: define **Views** $\rightarrow$ derive **Capabilities** $\rightarrow$ state **goal formulas** $\rightarrow$ prove **minimal compromise sets** $\rightarrow$ map to levels $L_{\mathrm{CORR}}, L_{\mathrm{PRIV}}, L_{\mathrm{LIVE}}$.
- **Step 3: Finish the property-by-property arguments.**
  - for each variant: short CORR/PRIV/LIVE argument aligned with the CRE abstraction (capability level).
- **Step 4: Fill the CRE results table with coding-sheet-backed entries.**
  - every table cell must be traceable to a concrete minimal-compromise argument in the coding sheet.
- **Suggested order (fast momentum):** $s_{\mathsf{ivot}}(p_2) \rightarrow s_{\mathsf{ivot}}(p_3) \rightarrow s_{\mathsf{iot}}(p_2) \rightarrow s_{\mathsf{mixer}}(p_3 \text{ or } p_4) \rightarrow s_{\mathsf{anonser}}(p_3) \rightarrow s_{\mathsf{ppml}}(p_3)$.

Future work

- Applications to other fields: cybersecurity, forensics, criminal justice, law, anywhere trust and corruption are at stake.
- A double-edged sword: the methodology provides insights for both defenders and attackers.

Dziękuję!

References I

[1] S. Barański, J. Szymański, A. Sobecki, D. Gil, and H. Mora, "Practical I-Voting on Stellar Blockchain," *Applied Sciences*, vol. 10, no. 21, p. 7606, Oct. 2020, ISSN: 2076-3417. DOI: 10.3390/app10217606 Accessed: Jul. 19, 2022.

[2] A. Sobecki, S. Barański, and J. Szymański, "Privacy-Preserving, Scalable Blockchain-Based Solution for Monitoring Industrial Infrastructure in the Near Real-Time," *Applied Sciences*, vol. 12, no. 14, p. 7143, Jul. 2022, ISSN: 2076-3417. DOI: 10.3390/app12147143 Accessed: Jul. 19, 2022.

[3] S. Baranski, M. Dotan, A. Lotem, and M. Vald, *Haze and Daze: Compliant Privacy Mixers*, 2023. Accessed: Oct. 15, 2024.

[4] S. Barański, J. Szymański, and H. Mora, "Anonymous provision of privacy-sensitive services using blockchain and decentralised storage," *International Journal of Information Security*, vol. 24, no. 3, p. 130, May 2025, ISSN: 1615-5270. DOI: 10.1007/s10207-025-01052-w Accessed: Jun. 19, 2025.

[5] S. Baranski and J. Szymanski, *Federated Distributed Key Generation*, Feb. 2025. DOI: 10.48550/arXiv.2502.20835 arXiv: 2502.20835 [cs]. Accessed: Jun. 19, 2025.

[6] S. Barański, "Nieświadome sieci neuronowe," in *Algorytmy i zastosowania inteligencji obliczeniowej*, J. Szymański, Ed., Gdańsk: Wydawnictwo Politechniki Gdańskiej, 2022, ISBN: 978-83-7348-882-3.

References II

[7] S. Baranski, “A Survey on Privacy-Preserving Machine Learning Inference,” *TASK Quarterly*, vol. 28, no. 2, 2025.