Fraunhofer AISEC

A Case Study of Real-World Post-Quantum Migration

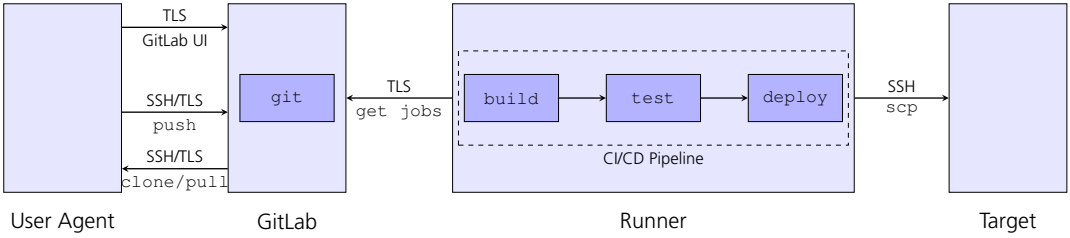
Migrating GitLab CI/CO DevOps Pipelines – or What's Left of the Original Problem

Daniel Loebenberger, European Conference on PQC Migration

Den Haag, December 03, 2025

Motivation

Post-Quantum Secure CI/CD DevOps Pipelines (Project AMiQuaSy 2023–2026)



Plan: Simple GitLab migration

genua.

XITASO

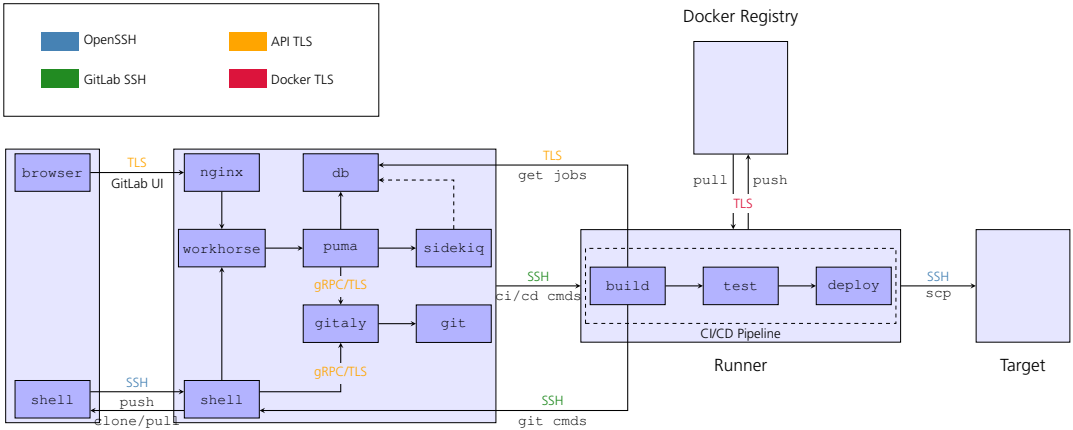
SPONSORED BY THE

Federal Ministry of Education and Research

OTH Amberg-Weiden

Motivation

Post-Quantum Secure CI/CD DevOps Pipelines (Project AMiQuaSy 2023–2026)

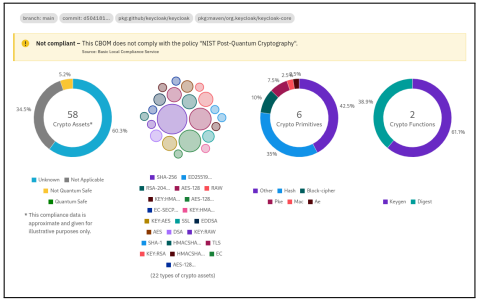


Reality: Many different (proprietary) subsystems

CBOMs to the Rescue

Semi-automated Generation of a Cryptographic Bill of Material

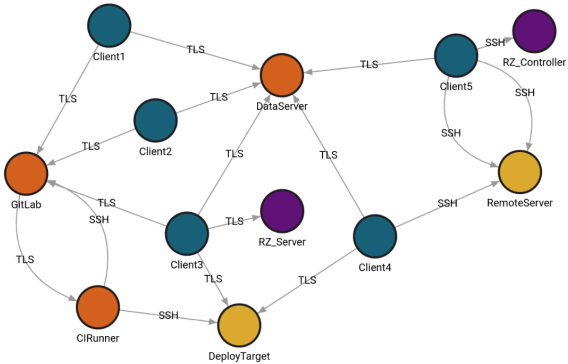
- CBOMs as special cases of SBOMs
- Tooling available, e.g. CycloneDX
- Suitable standardized data-formats (`json`)
- Collection of crypto on systems
- Proper visualization
- Manual assessment still often necessary
- LLMs facilitate analysis of human-readable descriptions of packet managers
- However, asset collection in large infrastructures is an unsolved problem!



cf. Hirsch, E., Raab, K., Bauer, T. J., & Loebenberger, D. (2025). Detecting Cryptographically Relevant Software Packages with Collaborative LLMs

CBOMs to the Rescue

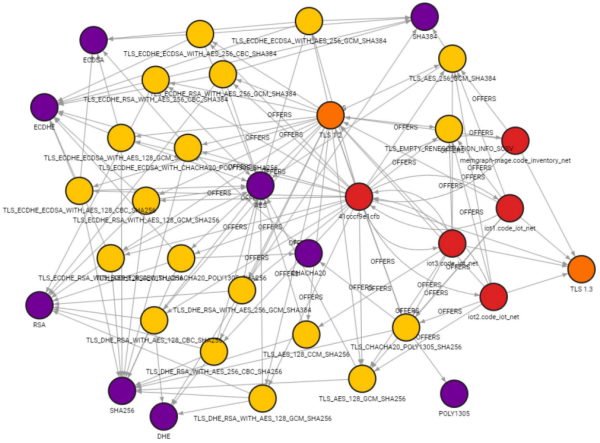
Semi-automated Generation of a Cryptographic Bill of Material



Store topological information from Wireshark / `pcap`
in a graph database such as `memgraph`

Full Graph

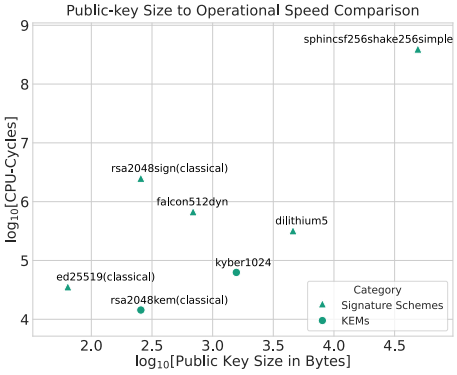
including all collected migration information



(qualitative figure)

Replacement of Cryptographic Algorithms

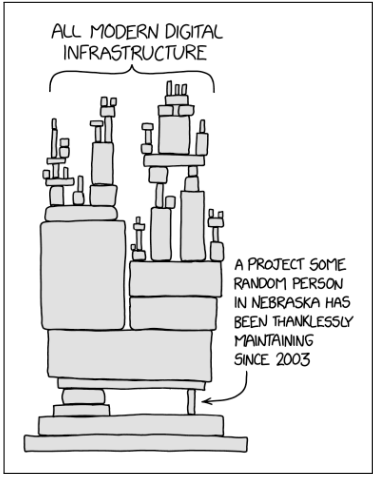
- Identification of affected algorithms
- Examination of the APIs of the crypto libraries
- Analysis of the data formats used
- Determining the calling OS and application code
- Determining the called OS and application code
- Quantitative description of algorithm features
- Identification of algorithmic dependencies
- Assessment of new trade-offs
- Possible impact of hybrid mechanisms



Data: `supercop`

Replacement of the Protocols Themselves

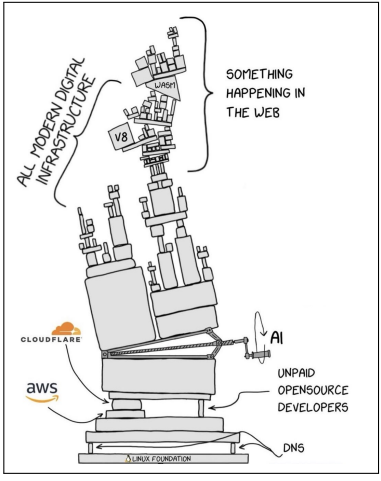
- Negotiation of cryptographic procedures
- Handshake protocols for key exchange
- Invocation of cryptographic procedures
- Current key sizes and hardware/software limits
- Thresholds for latency and throughput
- Sources of keys and certificates
- Possible use of cryptographic hardware
- . . .



Source: https://xkcd.com/2347/

Replacement of the Protocols Themselves

- Negotiation of cryptographic procedures
- Handshake protocols for key exchange
- Invocation of cryptographic procedures
- Current key sizes and hardware/software limits
- Thresholds for latency and throughput
- Sources of keys and certificates
- Possible use of cryptographic hardware
- . . .



Source: Modern interpretation by Equivalent_Site6616

Remaining Operational Problems

since Ott et al. (2019)

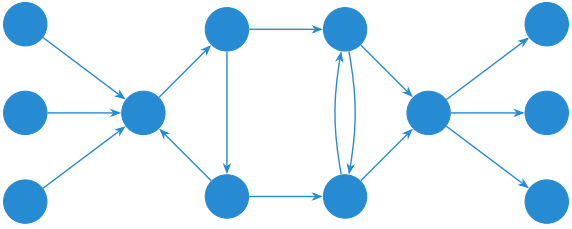
There is no structured approach to cryptographic migration: the approaches for migration are always some kind of (guided) best-practice tasks

cf. Näther, C., Herzinger, D., Gazdag, S.-L., Steghöfer, J.-P., Daum, S., & Loebenberger, D. (2024). Migrating Software Systems Toward Post-Quantum Cryptography
Näther, C., Herzinger, D., Steghöfer, J.-P., Gazdag, S.-L., Hirsch, E., & Loebenberger, D. (2024). SoK: Towards a Common Understanding of Cryptographic Agility

»We build our computers the way we build our cities – over time, without a plan, on top of ruins.«
(Ellen Ullman)

Definition (Migration Cluster aka. Strongly Connected Component)

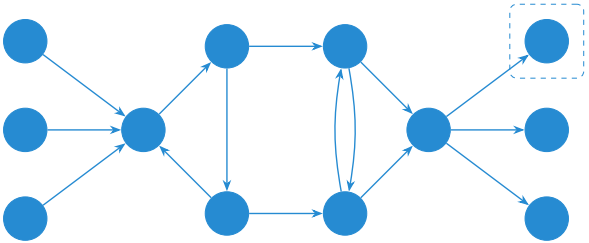
Let $G = (V, E)$ be a migration graph and $v \in V$. The *migration cluster* $c(v)$ *of* v is the set of all components $w \in V$ such that w is in the set $\text{dep}(v)$ of all dependencies of v and vice versa $v \in \text{dep}(w)$. The migration cluster $c(v)$ is exactly the set of components that have to be migrated together with $v \in V$ due to mutual dependencies.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Definition (Migration Cluster aka. Strongly Connected Component)

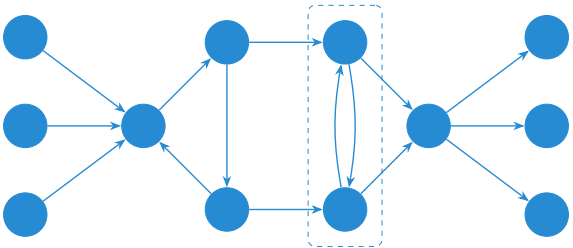
Let $G = (V, E)$ be a migration graph and $v \in V$. The *migration cluster* $c(v)$ *of* v is the set of all components $w \in V$ such that w is in the set $\text{dep}(v)$ of all dependencies of v and vice versa $v \in \text{dep}(w)$. The migration cluster $c(v)$ is exactly the set of components that have to be migrated together with $v \in V$ due to mutual dependencies.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Definition (Migration Cluster aka. Strongly Connected Component)

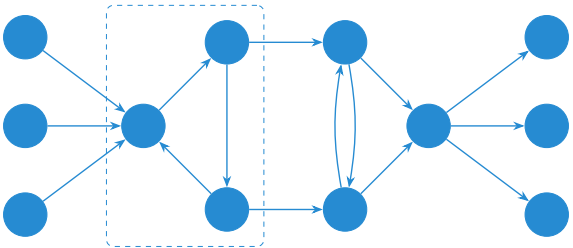
Let $G = (V, E)$ be a migration graph and $v \in V$. The *migration cluster* $c(v)$ *of* v is the set of all components $w \in V$ such that w is in the set $\text{dep}(v)$ of all dependencies of v and vice versa $v \in \text{dep}(w)$. The migration cluster $c(v)$ is exactly the set of components that have to be migrated together with $v \in V$ due to mutual dependencies.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Definition (Migration Cluster aka. Strongly Connected Component)

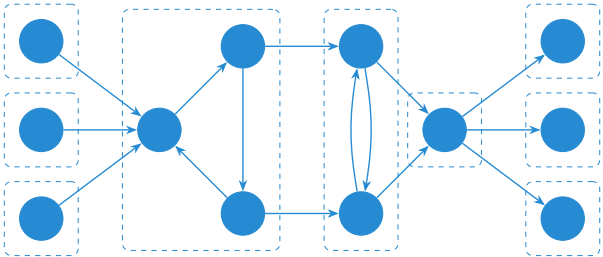
Let $G = (V, E)$ be a migration graph and $v \in V$. The *migration cluster* $c(v)$ *of* v is the set of all components $w \in V$ such that w is in the set $\text{dep}(v)$ of all dependencies of v and vice versa $v \in \text{dep}(w)$. The migration cluster $c(v)$ is exactly the set of components that have to be migrated together with $v \in V$ due to mutual dependencies.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Definition (Migration Cluster aka. Strongly Connected Component)

Let $G = (V, E)$ be a migration graph and $v \in V$. The *migration cluster* $c(v)$ *of* v is the set of all components $w \in V$ such that w is in the set $\text{dep}(v)$ of all dependencies of v and vice versa $v \in \text{dep}(w)$. The migration cluster $c(v)$ is exactly the set of components that have to be migrated together with $v \in V$ due to mutual dependencies.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

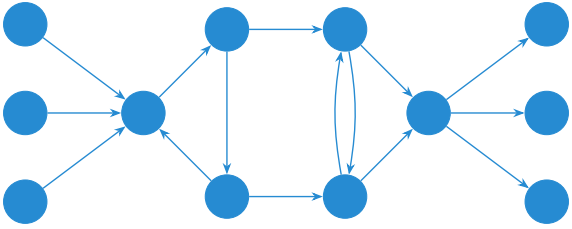
Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

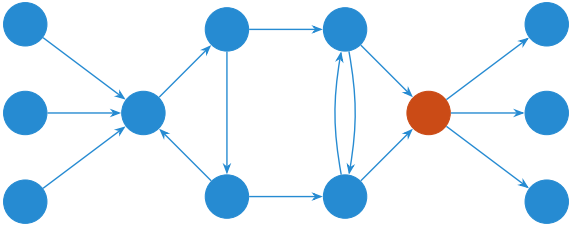
Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

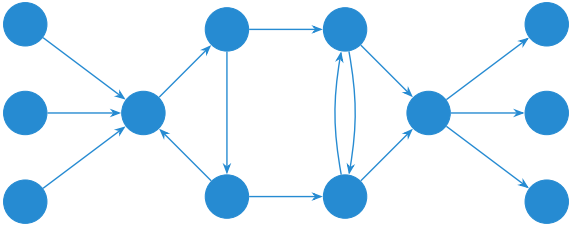
Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

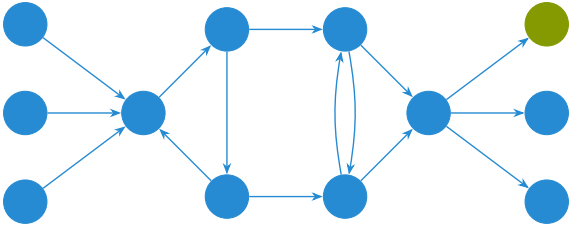
Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

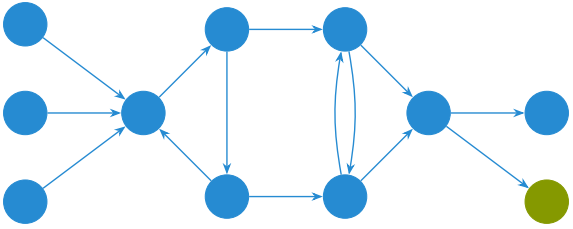
Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

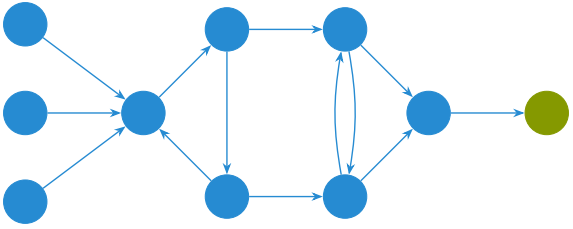
Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

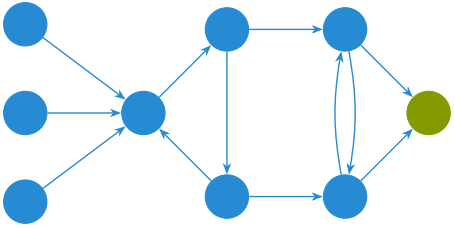
Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

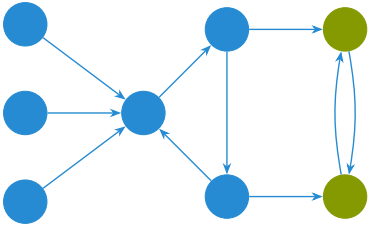
Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

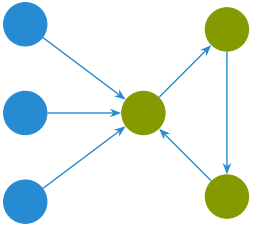
Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.



cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.

cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.

cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.

cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Formalization of the Migration Problem

Migratable Components and Migration Strategies

The set of *migratable components around* v is given by

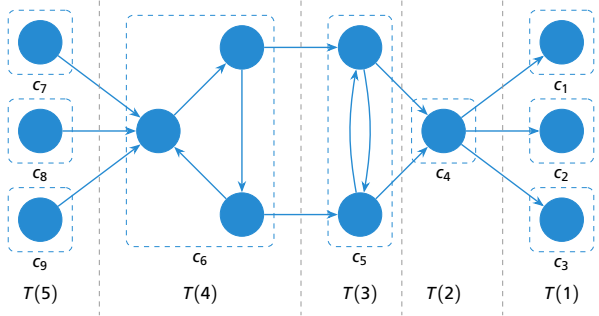
$$m(v) = \begin{cases} c(v) & \text{, if } c(v) \to \bar{c}(v) = \emptyset \\ \emptyset & \text{, otherwise} \end{cases}$$

where $\bar{c}(v) := V \setminus c(v)$.

cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Formalization of the Migration Problem

Overall Structure of Migration Graphs



- G partitions into the migration clusters c of its components
- The migration clusters have acyclic dependencies between each other
- They have thus a canonical position $T(i)$ in the migration process

cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Final Hardness Result

Expectedly, we have:

- Dependencies suitably bounded $\Longrightarrow$ large depth $d(G)$
- In randomly sampled migration projects we have $\sim n/\log n$ migration clusters
- The migration graph decomposes in $\sim n/\log n$ migration clusters
- The clusters have size $\sim \log n$ with standard deviation $\sim \sqrt{n}/\log n$
- The condensation graph is connected and thus not sparse

cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Final Hardness Result

Expectedly, we have:

- Dependencies suitably bounded $\Longrightarrow$ large depth $d(G)$
- In randomly sampled migration projects we have $\sim n/\log n$ migration clusters
- The migration graph decomposes in $\sim n/\log n$ migration clusters
- The clusters have size $\sim \log n$ with standard deviation $\sim \sqrt{n}/\log n$
- The condensation graph is connected and thus not sparse

Conclusion

Any sufficiently large migration project stemming from the model above has an intrinsic complexity due to many dependent (comparatively small) migration clusters. The migration of these clusters either takes many successive steps or includes at least one particularly difficult one.

cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Final Hardness Result

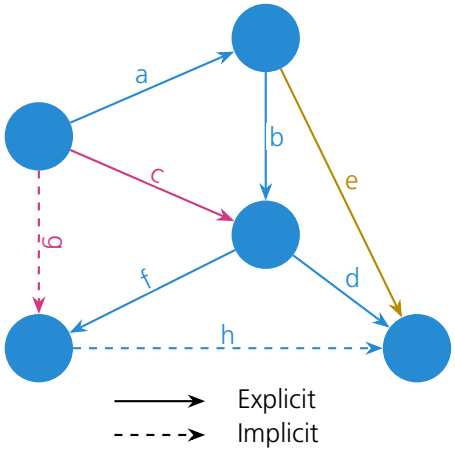
Numerical Evaluation

$\#V$	$\mathrm{E}(\#c(v))$	$\mathrm{E}(\#(G/c))$	$\mathrm{SD}(\#(G/c))$	$d(G)$	
n	$\log n$	$\frac{n}{\log n}$	$\frac{\sqrt{n}}{\log n}$	$\frac{n}{\log^2 n}$	$\sqrt{n}$
10	2	4	1	2	3
100	5	22	2	5	10
1 000	7	145	5	21	32
10 000	9	1 086	11	118	100
100 000	12	8 686	27	754	316
1 000 000	14	72 382	72	5 239	1 000
10 000 000	16	620 421	196	38 492	3 162
100 000 000	18	5 428 681	543	294 706	10 000
1 000 000 000	21	48 254 942	1 526	2 328 539	31 623

cf. Loebenberger, D., Gazdag, S.-L., Herzinger, D., Hirsch, E., Näther, C., & Steghöfer, J.-P. (2024). On the Formalization of Cryptographic Migration

Extension: Dependency Graphs with Implicit Dependencies

- How to get the graph from real-world infrastructures?
- Start with explicit dependencies (***a***, ***b***, ***d***, ***f***)
- Successively get rid of irrelevant or redundant dependencies (***c***, ***e***, ***g***)
- Iteratively add implicit dependencies (***h***)
- Employ a functional predicate to refine the model
- Repeat until model seems complete



cf. Nzetchuen, E., Igler, B., Loebenberger, D., & Stöttinger, M. (2025). Cryptographic Migration with Implicit Dependencies. Work in progress.

PQC Migration Workshop at Eurocrypt 2026 in Rome

Submission: February 01, 2026

MAgiCS Workshop 2026 About Call for Papers Program Committee Venue Organizers

Migration and Agility in Cryptographic Systems

Co-located with **Eurocrypt 2026**. as an affiliated workshop.
Date: 10 May 2026 · **Location:** Città Universitaria, Sapienza University of Rome

The Workshop on Migration and Agility in Cryptographic Systems (MAgiCS) will take place on the 10th of May 2026 at the Città Universitaria (University Campus) of Sapienza University of Rome, co-located with Eurocrypt 2026. MAgiCS 2026 focuses on the topic of migration and the transition of cryptographic systems. The workshop aims to bridge the gap between theoretical concepts and practical processes for their application in real-world scenarios.

Last updated: 6 Nov 2025

About

Abstract

Cryptographic migration, specifically in the post-quantum setting, is a challenging and, in practice, mainly unsolved

Fraunhofer AISEC

Contact

Prof. Dr. Daniel Loebenberger
daniel.loebenberger@aisec.fraunhofer.de

Fraunhofer Institut für
Angewandte und Integrierte Sicherheit AISEC
Hermann-Brenner-Platz 1
92637 Weiden i.d.Opf.