Fraunhofer AISEC

TRANSFER for Our FUTURE

A Formal Model and Lower-Bound Intuition for Cryptographic Migration

Daniel Loebenberger, Eduard Hirsch, Stefan-Lukas Gazdag, Daniel Herzinger, Christian Näther, and Jan-Philipp Steghöfer

MAgiCS 2026 @ Sapienza Università di Roma, May 10, 2026

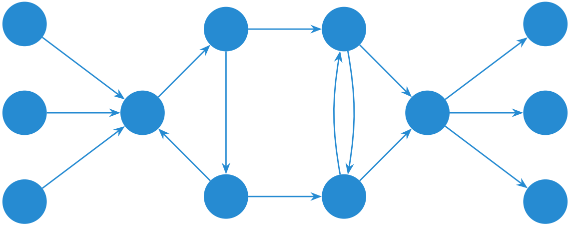
Motivation

All (non-trivial) migration projects appear
– at least empirically –
to be intrinsically complex.

What can we formally prove about this?

Definition (Migration Cluster aka. Strongly Connected Component)

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

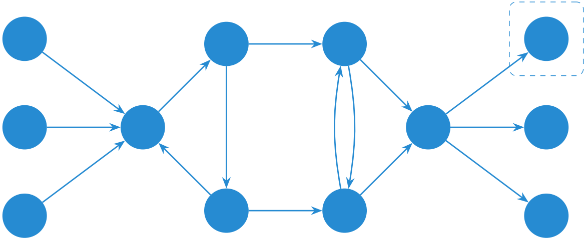


  Offen Fraunhofer AISEC

Formalization of the Migration Problem
Migration Graphs

Definition (Migration Cluster aka. Strongly Connected Component)

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

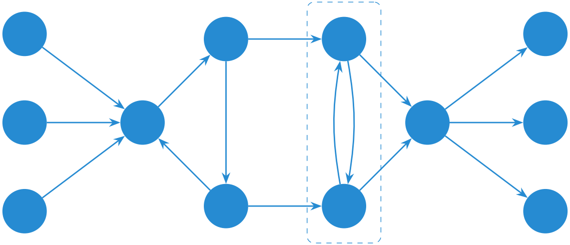


  Offen Fraunhofer AISEC

Formalization of the Migration Problem
Migration Graphs

Definition (Migration Cluster aka. Strongly Connected Component)

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

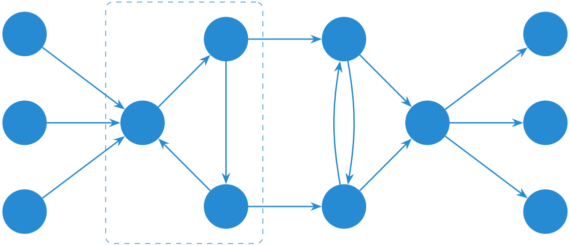


Formalization of the Migration Problem

Migration Graphs

Definition (Migration Cluster aka. Strongly Connected Component)

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

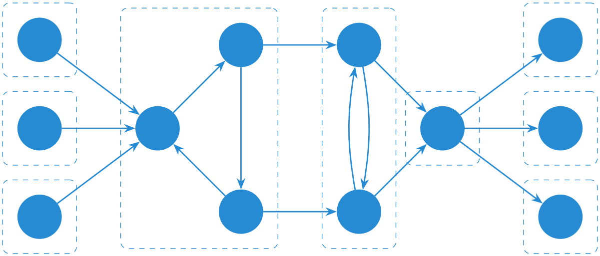


  Offen

Formalization of the Migration Problem
Migration Graphs

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.

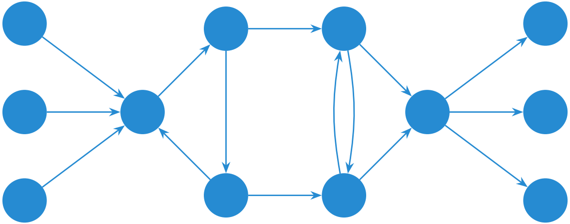


Formalization of the Migration Problem
Migratable Components and Migration Strategies

The set of *migratable components around* $\boldsymbol{v}$ is given by

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

where $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then $\boldsymbol{v}$ is *migratable*.



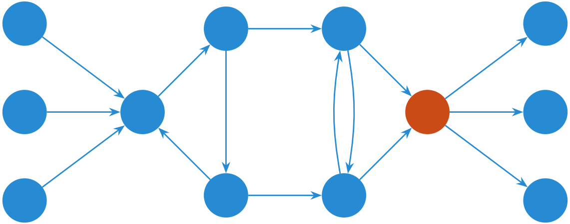
Formalization of the Migration Problem

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 $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then v is *migratable*.



  Offen

Fraunhofer AISEC

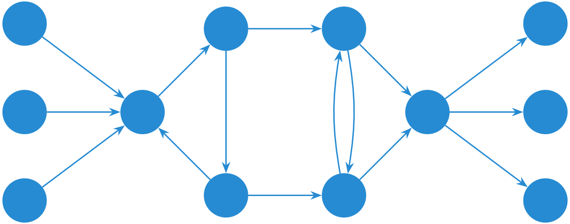
Formalization of the Migration Problem

Migratable Components and Migration Strategies

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

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

where $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then v is *migratable*.

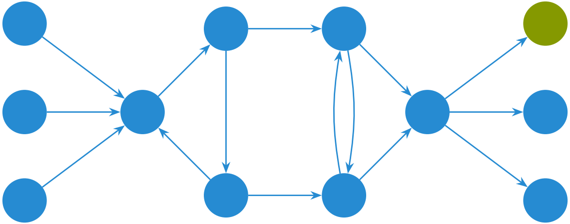


Formalization of the Migration Problem
Migratable Components and Migration Strategies

The set of *migratable components around* $\boldsymbol{v}$ is given by

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

where $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then $\boldsymbol{v}$ is *migratable*.



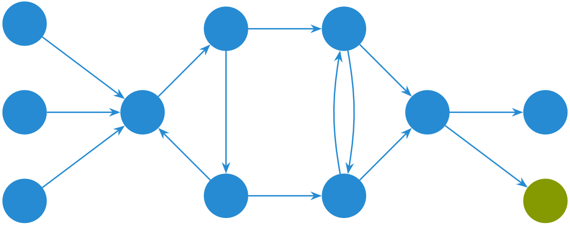
Formalization of the Migration Problem

Migratable Components and Migration Strategies

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

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

where $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then v is *migratable*.



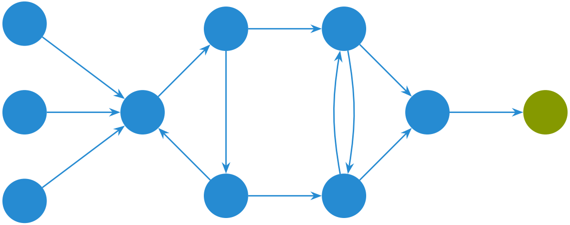
  Offen

Formalization of the Migration Problem
Migratable Components and Migration Strategies

The set of *migratable components around* $\boldsymbol{v}$ is given by

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

where $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then $\boldsymbol{v}$ is *migratable*.



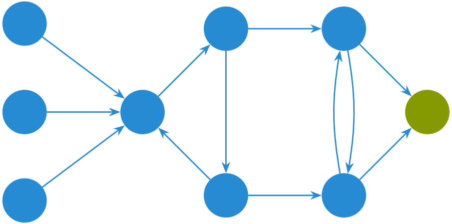
  Offen

Formalization of the Migration Problem
Migratable Components and Migration Strategies

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

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

where $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then v is *migratable*.



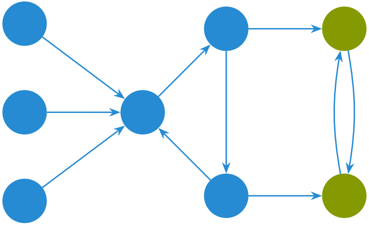
Formalization of the Migration Problem

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 $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then v is *migratable*.

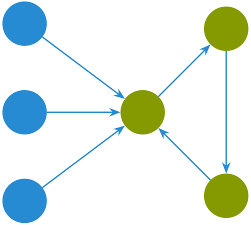


Formalization of the Migration Problem
Migratable Components and Migration Strategies

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 $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then v is *migratable*.



  Offen

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 $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then v is *migratable*.

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 $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then v is *migratable*.

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 $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then v is *migratable*.

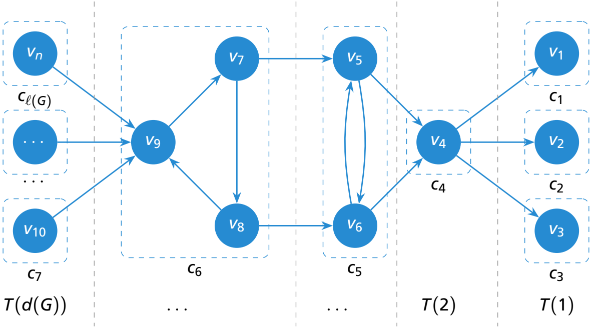
Formalization of the Migration Problem
Migratable Components and Migration Strategies

The set of *migratable components around* $\boldsymbol{v}$ is given by

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

where $\bar{c}(v) := V \setminus c(v)$. If $m(v) \neq \emptyset$ then $\boldsymbol{v}$ is *migratable*.

Formalization of the Migration Problem
Overall Structure of Migration Graphs

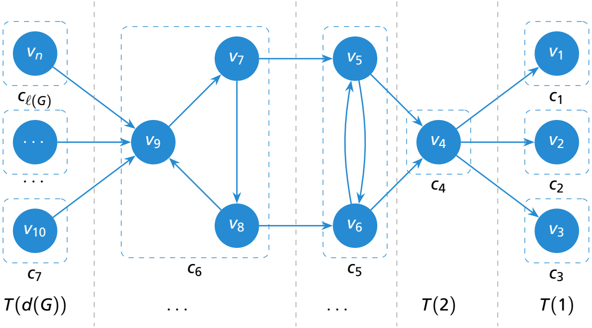


- G partitions into ℓ *migration clusters* $\mathbf{c}$ of its components
- The migration clusters have acyclic dependencies between each other
- They have thus a *canonical position* $T(i)$ in the migration process

  Offen

Formalization of the Migration Problem

Overall Structure of Migration Graphs



- We can thus form the *acyclic condensation* $\boldsymbol{G}/\boldsymbol{c}$ of $\boldsymbol{G}$ mod $\boldsymbol{c}$
- A *migration strategy* is a sequence of migrateble components which covers all clusters
- Theorem: All strategies have the same length $\ell(\boldsymbol{G}) = \#(\boldsymbol{G}/\boldsymbol{c})$

  Offen

Fraunhofer AISEC

Random Migration Graphs

- Use random Erdős-Rényi graphs on n vertices with connection probability p.

Theorem (A variant of Theorem 1 in Erdős & Rényi 1959)

Let G_{n,p_n} be a random directed n-vertex graph, where each arc is present with probability $p_n = \frac{\log n + c_n}{n}$. Then, for growing n, the probability that G is connected is given by

$$\lim_{n\to\infty} \operatorname{prob}(G_{n,p_n} \textit{ connected}) = \begin{cases} 0 & \text{, if } c_n \to -\infty \\ e^{-e^{-c}} & \text{, if } c_n \to c \text{ const.} \\ 1 & \text{, if } c_n \to \infty \end{cases}$$

Random Migration Graphs

- ~~Use random Erdős-Rényi graphs on n vertices with connection probability p.~~
- Consider random partitions on n vertices!

Random Migration Graphs

- ~~Use random Erdős-Rényi graphs on n vertices with connection probability p.~~
- Consider random partitions on n vertices!
- There are B_n different possible partitions of n components in disjoint subsets, where

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^n}{k!}$$

  is the nth Bell number and $B_0 = 1$.
- Theorem: One can sample random partitions efficiently. However, due to the involved probabilities the resulting experiments did turn out to be numerically unstable.

  Offen

Fraunhofer AISEC

Random Migration Graphs

- ~~Use random Erdős-Rényi graphs on n vertices with connection probability p.~~
- Consider random partitions on n vertices!
- There are B_n different possible partitions of n components in disjoint subsets, where

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^n}{k!}$$

  is the nth Bell number and $B_0 = 1$.
- Theorem: One can sample random partitions efficiently. However, due to the involved probabilities the resulting experiments did turn out to be numerically unstable.
- Let M with values in $\mathbb{N}_{\geq 1}$ be the number of partitions in a random partition sample Π.
- There are classical results on the behavior of Π and the behavior of M.

Random Migration Graphs

Expected number of migration clusters and expected migration cluster size

Theorem (Summary of Odlyzko & Richmond 1985)

The number of migration clusters M is expectedly $\mathrm{E}(M) \sim n/\log n$ and the expected size of each migration cluster is $\log n$. More specifically the distribution of the number of migration clusters M converges asymptotically to a normal distribution with mean $\mu = n/\log n$ and standard deviation $\sigma = \sqrt{n}/\log n$ for $n \to \infty$.

  Offen

Fraunhofer AISEC

Random Migration Graphs

Expected number of migration clusters and expected migration cluster size

Theorem (Summary of Odlyzko & Richmond 1985)

The number of migration clusters M is expectedly $\mathrm{E}(M) \sim n/\log n$ and the expected size of each migration cluster is $\log n$. More specifically the distribution of the number of migration clusters M converges asymptotically to a normal distribution with mean $\mu = n/\log n$ and standard deviation $\sigma = \sqrt{n}/\log n$ for $n \to \infty$.

Lemma

The expected maximal number of migration clusters when looking independently at s randomly sampled migration projects is bounded above by

$$\frac{n}{\log n} + \frac{\sqrt{2n \log s}}{\log n}.$$

$\Rightarrow$ We would have to sample exponentially many migration graphs (in n) to expectedly observe graphs with asymptotically more than $n/\log n$ migration clusters.

Properties of the Condensation Graph

Again Connectedness

Theorem (A variant of Theorem 1 in Erdős & Rényi 1959)

Let G_{n,p_n} be a random directed n-vertex graph, where each arc is present with probability $p_n = \frac{\log n + c_n}{n}$. Then, for growing n, the probability that G is connected is given by

$$\lim_{n\to\infty} \operatorname{prob}(G_{n,p_n} \text{ connected}) = \begin{cases} 0 & \text{, if } c_n \to -\infty \\ e^{-e^{-c}} & \text{, if } c_n \to c \text{ const.} \\ 1 & \text{, if } c_n \to \infty \end{cases}$$

$\Rightarrow$ Even in the sparsest case, we expect a considerable number of dependencies between the migration clusters!

Properties of the Condensation Graph

Depth-Width Trade-off – Independent of the Orientation Chosen

Theorem

Let $w : \mathbb{R} \to \mathbb{R}$ *be a smooth function in a real variable with* $w \in o(n')$*, e. g.,* $w(n') = \log n'$*. Assume for any cluster* $c(v)$ *in a condensation graph* G/c *with* $n' \sim \frac{n}{\log n}$ *components, we have for each* $k \geq 0$ *the bound* $\#(c(v) \to^k V/c) \leq w(n')$*. Then* $d(G) \geq n'/w(n')$*. In other words, the width of the migration graph has a trade-off with the depth of the migration graph.*

  Offen

Fraunhofer AISEC

Final Hardness Result
Numerical Evaluation

$\#V$	$\mathrm{E}(\#c(v))$	$\mathrm{E}(\#(G/c))$	$\mathrm{SD}(\#(G/c))$	$d(G)$	
n	$\log n$	$\frac{n}{\log n}$	$\frac{\sqrt{n}}{\log n}$	$\frac{n}{\log^2 n}$ for $w(n') = \log n'$	$\sqrt{n}$ for $w(n') = \sqrt{n'} / \log n'$
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

Final Hardness Result

Summary

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

Final Hardness Result
Summary

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.

Remaining Operational Problems
since Ott et al. (2019)

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

Contact

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

Fraunhofer Institute for
Applied and Integrated Security AISEC
Hermann-Brenner-Platz 1
92637 Weiden i.d.Opf.