“That’s my perspective from 30 years of doing this”

An Interview Study on Practices, Experiences, and Challenges of Updating Cryptographic Code

Alexander Krause, Harjot Kaur, Jan H. Klemmer, Oliver Wiese, and Sascha Fahl

#teamusec

CISPA Helmholtz Center for Information Security, Hannover, Germany

Why Crypto Updates Matter

- Cryptographic code loses security guarantees over time
- SHA-1 deprecation took over a decade
- Post-quantum cryptography transition underway
- Previous work shows many products contain outdated crypto

Why Crypto Updates Matter

- Cryptographic code loses security guarantees over time
- SHA-1 deprecation took over a decade
- Post-quantum cryptography transition underway
- Previous work shows many products contain outdated crypto

"Never implement your own crypto" - but what about updating it?

Research Approach

Research Questions

- RQ1: How do developers become aware of crypto updates?
- RQ2: Why do developers update crypto implementations?
- RQ3: What are the update processes?
- RQ4: What challenges do developers face?

Methodology

- 21 semi-structured interviews
- Experienced developers with crypto update experience
- Mixed recruitment: professional contacts, email, Upwork
- Participants from companies, open-source projects, freelancers

Study Participants (N=21)

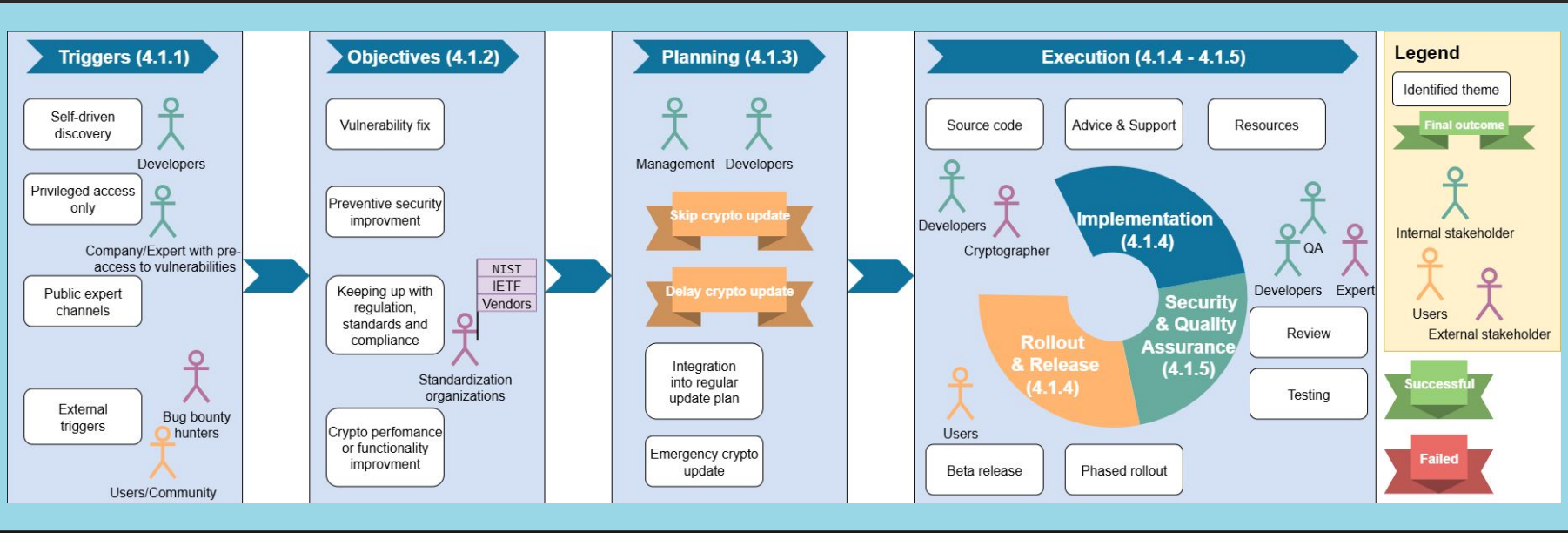
Demographics

- Experience: 1-35 years development, 0-37 years security
- Roles: Software engineers, CTOs, researchers, students
- Projects: Solo projects, companies, open-source
- Education: High school to PhD

Participants

- Group S (13): Self-implemented crypto (libraries, primitives)
- Group U (12): Used existing crypto implementations
- 4 participants in both groups

Six-Phase Update Process



13 Identified Challenges

Categorized by Phase

- **Awareness:** Staying up to date, lack of information
- **Planning:** Changing standards, backward compatibility, legacy support, lack of structured processes
- **Implementation:** Understanding crypto, lack of expertise, documentation issues
- **Quality:** Trust in third-party crypto
- **Rollout:** Technical issues, UI changes, performance problems

13 Identified Challenges

Categorized by Phase

- **Awareness:** Staying up to date, lack of information
- **Planning:** Changing standards, backward compatibility, legacy support, lack of structured processes
- **Implementation:** Understanding crypto, lack of expertise, documentation issues
- **Quality:** Trust in third-party crypto
- **Rollout:** Technical issues, UI changes, performance problems

> “One problem is that a lot of the stuff is used in embedded systems and basically systems that don’t change much. You’ve got 10 or 15-year-old implementations.”— P18

Crypto Updates vs. Regular Software Updates

- **Criticality:** Data breach vs. mispositioned logo
- **Review Requirements:** Six-eye rule vs. standard review
- **Complexity:** Algorithm understanding vs. feature implementation
- **Legacy Impact:** Embedded systems with 10-15 year lifespans
- **Expertise Required:** Specialized crypto knowledge vs. general development

Crypto Updates vs. Regular Software Updates

- **Criticality:** Data breach vs. mispositioned logo
- **Review Requirements:** Six-eye rule vs. standard review
- **Complexity:** Algorithm understanding vs. feature implementation
- **Legacy Impact:** Embedded systems with 10-15 year lifespans
- **Expertise Required:** Specialized crypto knowledge vs. general development

"If you don't update your PDF library, you may have a logo positioned wrongly. But if you don't update a crypto library, you may have your data leaked." — P13

Real-World Examples

- "Following the right people" — P3

- “It has taken ten years. However, at that time, because the project was mostly an open-source side project, for sure, there was no full-time work on this. Maybe it’s half a year in total.”— P9

What Developers Want

Key Desires

- **Education**: Crypto onboarding, continuous learning
- **Expertise**: Dedicated crypto team members
- **Resources**: Better documentation, practical examples
- **Tools**: Improved APIs, automated testing
- **Processes**: Clear best practices, structured approaches

What Developers Want

Key Desires

- **Education**: Crypto onboarding, continuous learning
- **Expertise**: Dedicated crypto team members
- **Resources**: Better documentation, practical examples
- **Tools**: Improved APIs, automated testing
- **Processes**: Clear best practices, structured approaches

> "We didn't have any real best practices regarding security audits or security updates." — P7

Our Recommendations

For Developers

- Consider crypto updates in initial design
- Follow crypto experts on social media/blogs
- Treat crypto as permanent technical debt
- Plan for regular crypto reviews

For Library Developers

- Provide secure defaults and migration paths
- Document changes thoroughly
- Consider automated update tools

For Standards Organizations

- Provide actionable change logs
- Include implementation examples
- Simplify explanations for developers

Our Recommendations

For Developers

- Consider crypto updates in initial design
- Follow crypto experts on social media/blogs
- Treat crypto as permanent technical debt
- Plan for regular crypto reviews

For Library Developers

- Provide secure defaults and migration paths
- Document changes thoroughly
- Consider automated update tools

For Standards Organizations

- Provide actionable change logs
- Include implementation examples
- Simplify explanations for developers

Our Recommendations

For Developers

- Consider crypto updates in initial design
- Follow crypto experts on social media/blogs
- Treat crypto as permanent technical debt
- Plan for regular crypto reviews

For Library Developers

- Provide secure defaults and migration paths
- Document changes thoroughly
- Consider automated update tools

For Standards Organizations

- Provide actionable change logs
- Include implementation examples
- Simplify explanations for developers

Our Recommendations

For Developers

- Consider crypto updates in initial design
- Follow crypto experts on social media/blogs
- Treat crypto as permanent technical debt
- Plan for regular crypto reviews

For Library Developers

- Provide secure defaults and migration paths
- Document changes thoroughly
- Consider automated update tools

For Standards Organizations

- Provide actionable change logs
- Include implementation examples
- Simplify explanations for developers

Preparing for the PQC Transition

- Largest cryptographic transition in history
- Limited experience with PQC implementations
- Multiple updates expected as standards evolve
- Our findings provide crucial insights for this transition

Apply our recommendations proactively

Key Takeaways

Main Findings

- Crypto updates are complex and unique
- Developers lack structured processes
- Backward compatibility is the major challenge
- Better support tools and processes are needed

Future Work

- Automated crypto update detection
- Developer tooling improvements
- Best practices development

Alexander Krause
CISPA Helmholtz Center for Information Security
Hannover, Germany

alexander.krause@cispa.de

@akrause_de

https://akrause.de