# "We make it a big deal in the company": Security Mindsets in Organizations that Develop Cryptographic Products

Julie M. Haney[1], Mary F. Theofanos[1], Yasemin Acar[2], Sandra Spickard Prettyman[3]

[1]National Institute of Standards and Technology
{julie.haney, mary.theofanos}@nist.gov

[2]Leibniz University Hannover
acar@sec.uni-hannover.de

[3]Culture Catalyst
sspretty50@icloud.com

## ABSTRACT

Cryptography is an essential component of modern computing. Unfortunately, implementing cryptography correctly is a non-trivial undertaking. Past studies have supported this observation by revealing a multitude of errors and developer pitfalls in the cryptographic implementations of software products. However, the emphasis of these studies was on individual developers; there is an obvious gap in more thoroughly understanding cryptographic development practices of organizations. To address this gap, we conducted 21 in-depth interviews of highly experienced individuals representing organizations that include cryptography in their products. Our findings suggest a security mindset not seen in other research results, demonstrated by strong organizational security culture and the deep expertise of those performing cryptographic development. This mindset, in turn, guides the careful selection of cryptographic resources and informs formal, rigorous development and testing practices. The enhanced understanding of organizational practices encourages additional research initiatives to explore variations in those implementing cryptography, which can aid in transferring lessons learned from more security-mature organizations to the broader development community through educational opportunities, tools, and other mechanisms. The findings also support past studies that suggest that the usability of cryptographic resources may be deficient, and provide additional suggestions for making these resources more accessible and usable to developers of varying skill levels.

## 1. INTRODUCTION

In a dynamic, threat-laden, and interconnected digital environment, cryptography protects privacy, provides for anonymity, ensures the confidentiality and integrity of communications, and safeguards sensitive information. Given the need for cryptography, there is an abundance of cryptographic algorithm and library choices for developers wishing to integrate cryptography into their products and services. However, developers often lack the expertise to navigate these choices, resulting in the introduction of security vulnerabilities [27]. A 2016 industry survey that included over 300,000 code assessments found that 39% of those applications had cryptographic problems [72]. Implementing cryptography correctly is a non-trivial undertaking.

In 1997, security expert Bruce Schneier commented on the lack of cryptographic implementation rigor and expertise at that time, asserting, "You can't make systems secure by tacking on cryptography as an afterthought. You have to know what you are doing every step of the way, from conception to installation" [61]. Past studies have supported this observation by revealing a multitude of errors in the cryptographic implementations of software products (e.g., [17–19, 42]) and the pitfalls developers encounter when including cryptography within products (e.g,. [1, 2, 48]). This body of research suggests that developers have not progressed much in the past 20 years. However, as these studies have been largely focused on individual practices outside the professional work context or on the development of mobile apps, it is unclear if these shortcomings also apply to organizational development and testing, particularly among organizations for which security and cryptography are essential components. One exploratory survey examined high-level organizational practices in cryptographic development, but lacked rich insight into actual practices and motivators behind those [31]. Clearly, there is a gap in the literature in more thoroughly understanding organizational cryptographic development practices.

To address this gap, we performed a qualitative investigation into the processes and resources that organizations employ to ensure their cryptographic products are not fraught with errors and vulnerabilities. We define the scope of cryptographic products as those implementing cryptographic algorithms or using crypto (cryptography) to perform some function. We conducted 21 in-depth interviews involving participants representing organizations that develop either a security product that uses cryptography or a non-security product that heavily relies on cryptography. Unlike previous studies, our participants were professionals who were highly experienced in cryptographic development and testing, not computer science students or developers with little cryptographic experience.

The study aimed to answer the following research questions:

**Q1** What are the cryptographic development and testing practices of organizations?

*USENIX Symposium on Usable Privacy and Security (SOUPS)* 2018.
August 12–14, 2018, Baltimore, MD, USA.

**Q2** What challenges, if any, do organizations encounter while developing and testing these products?

**Q3** What cryptographic resources do these organizations use, and what are their reasons for choosing these?

Our findings went deeper than uncovering practices, revealing a security mindset not noted in other research results. We discovered that some organizations believe they have achieved the expertise and rigor recommended by Schneier. Compared to developer populations studied in the past, the organizations in our investigation appear to have a stronger security culture and are more mature in their cryptography and security experience. The strong security culture we observed does not appear to be linked to company size or available resources. These security mindsets permeate the entire development process as they inform judicious selection of cryptographic resources and rigorous development practices.

Our work has several contributions. To our knowledge, this is the first in-depth study to explore cryptographic development practices and security mindsets in organizations from the viewpoint of those with extensive experience in the field. While some of the practices identified in our study may be considered known best practice within the security community, our paper is novel in that there are few research studies documenting occurrences of strong security culture and development in actual practice, and none within the cryptography context. Our study provides systematic, scientific validation to the anecdotal point often made by security experts that there is no magical, one-dimensional solution to cryptographic development. Rather, good crypto is the result of a concerted effort to build expertise and implement secure development practices. The enhanced understanding encourages additional research initiatives to explore variations in those implementing cryptography. This can aid in transferring lessons learned from more security-mature organizations to the broader development community through educational opportunities, tools, and other mechanisms. Our findings also support past studies that suggest that the usability of cryptographic resources may be deficient, and provide additional suggestions for making these resources more accessible and usable to developers of varying skill levels.

## 2. RELATED WORK

To provide context, this section begins with a brief overview of cryptographic standards and certifications frequently referenced in our interviews. We then underpin our assertion that cryptographic development is not a trivial undertaking by summarizing past research on crypto misuse and lack of crypto resource usability. We also present an overview of prior work on lack of security mindsets and secure development practices to serve as a contrast to the more security-conscious approaches of our study organizations.

### 2.1 Cryptographic Standards

Cryptographic algorithm standards are developed by consensus of community stakeholders (e.g. vendors, researchers, governments) to foster compatibility, interoperability, and minimum levels of security. These can be found in formal standards documents from organizations such as Institute of Electrical and Electronics Engineers (IEEE) [35], International Organization for Standardization (ISO) [36], and the U.S. National Institute of Standards and Technology (NIST) [52]. Likely due to the U.S. locations of most of the study organizations, the participants most often mentioned cryptographic requirements issued by NIST. As perhaps the best known government standard, the Federal Information Processing Standards Publication (FIPS) 140-2 "specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information" [49]. These requirements are mandatory for cryptographic products purchased by the U.S. Government, but also are used voluntarily outside the government. There are two certification programs associated with FIPS 140-2 [50, 51]. Under these programs, vendors may submit cryptographic algorithm and module implementations for validation testing to accredited testing laboratories.

### 2.2 Cryptographic Misuse

Numerous studies have highlighted the difficulty developers have in correctly implementing cryptography. In 2002, Gutmann observed that security bugs were often introduced by software developers who did not understand the implications of their choices [30]. Nguyen showed that even open-source implementations under public scrutiny have cryptographic flaws [53]. Lazar performed a systematic study of 269 cryptographic vulnerabilities in the Common Vulnerabilities and Exposures (CVE) database, noting that 17% of bugs were in cryptographic libraries and the remaining 83% were in individual applications, usually due to cryptographic library misuses [40]. Georgiev et al. discovered rampant misuse of Secure Sockets Layer (SSL) in security-critical applications due to poorly designed application programming interfaces (APIs) [25]. Fahl et al. analyzed 13,500 Android apps and found that 8% were susceptible to man-in-the-middle attacks [19]. Using static analysis, Egele et al. found similar issues, observing that 88% of over 11,000 examined Android apps contained a significant error in their use of a cryptographic API [18]. Li et al. analyzed 98 apps from the Apple App Store and found 64 (65.3%) that contained cryptographic misuse flaws [42].

### 2.3 Usability of Cryptographic Resources

Usability is often neglected in cryptographic resources such as standards and libraries, resulting in complex solutions that provide little assistance to developers in making secure choices [27, 48]. Several research groups attempted to remedy this by developing tools, e.g., OpenCCE [4], Crypto-Assistant [23] and Crypto Misuse Analyzer [64], to guide developers in choosing and integrating appropriate cryptographic methods. Others proposed more usable cryptographic libraries. Forler et al. developed libadacrypt, a cryptographic library created to be "misuse-resistant" [20]. Bernstein et al. created the Networking and Cryptography library (NaCl) [8], a cross-platform cryptographic library designed to avoid errors found in widely used cryptographic libraries like OpenSSL [55]. Acar et al. conducted a usability study of cryptographic APIs that revealed that, in addition to usable interfaces, clear documentation with code samples and support for common cryptographic tasks were important in aiding developers [1].

### 2.4 Security Development and Mindset

There is much to learn from an examination of secure development and testing practices since even the best implemented cryptography can be subverted by the flawed implementation of another system component. McGraw advocated for security to be integrated into all aspects of the software development lifecycle [43]. Several documents, for example the Microsoft Security Development Lifecyle [34] and the System Security Engineering Capability Maturity Model (SSE-CMM) [44], formally define secure development practices. More recently, the Open Web Application Security Project (OWASP) Secure Software Development Lifecycle project is working towards providing guidelines for web and application developers [54]. However, none of these resources specifically mentions considerations for cryptography.

Not surprisingly, the implementation of formal secure development processes is not an easy task. A 2016 Veracode survey of over 350 developers indicated that organizations are prevented from fully implementing a secure development process due to a variety of challenges, including security testing causing product timeline delays, complexity in supporting legacy security processes, security standards and policies varying across the organization, and developers not consistently following secure coding practices [73]. Kanniah and Mahrin found that a variety of organizational, technical, and human factors affected implementation of secure software development practices [38]. These factors included developer skill and expertise, communication among stakeholders, and collaboration between security experts and developers.

Failures in development and testing leading to security errors appear to reflect a deficiency in a security mindset. Schneier claimed that "Security requires a particular mindset. Security professionals – at least the good ones – see the world differently" [62]. A security mindset involves being able to think like an attacker, maintaining a commitment to secure practices, and perpetuating a strong security culture.

A need for a security mindset is revealed in several studies that explored reasons why developers make security errors. For example, Xie, Lipford, and Chu identified an absence of personal responsibility for security as well as a gap between developers' understanding of security and how to implement it [76]. Xiao et al. discovered that the failure to adopt secure development tools was heavily dependent on social environments and how tool information was communicated [75]. From a testing perspective, Potter and McGraw argued that security testing is commonly misunderstood and should be more risk-based, involving an understanding of a potential attacker's mentality [58]. Bonver and Cohen agreed, noting that security testers should work closely with architects and developers to identify potential vulnerabilities, taking into account how an attacker may exploit a system [9].

# 3. METHODOLOGY

Between January and June 2017, we conducted 21 interviews of individuals working in organizations that develop products that use cryptography. Following rigorous, commonly accepted qualitative research methods, we continued interviewing until we reached theoretical saturation, the point at which no new themes or ideas emerged from the data [45], exceeding the minimum of 12 interviews prescribed in qualitative research best practices [29].

Our research team was multidisciplinary and consisted of a computer scientist specializing in information security and human-computer interaction, a computer scientist specializing in usability, a mathematician with research experience in usable security and privacy, and a sociologist experienced in qualitative research. Having a diverse team may improve research quality "in terms of enabling sounder methodological design, increasing rigor, and encouraging richer conceptual analysis and interpretation" [6].

The study was approved by our institutional review board. Prior to the interviews, participants were informed of the purpose of the study and how their data would be used and protected. Interview data were collected and recorded without personal identifiers and not linked back to the participants or organizations. Interviews were assigned an identifier (e.g., C08) used for all associated data in the study.

## 3.1 Recruitment

To ensure that we could explore different perspectives within the cryptographic product space, our sampling frame consisted of individuals who had organizational experience designing, developing, or testing products that use cryptography or who were knowledgeable about and had played a key role in these activities (e.g., managers of teams that performed these tasks). We utilized a combination of purposeful and convenience sampling strategies, which are widely employed in exploratory qualitative research [56]. Purposeful sampling was used to select organizations of different sizes and participants who had knowledge and experience within this specialized topic area. This was combined with convenience sampling, where participants were sought based on ease of accessibility to the researchers and their willingness to participate in the study.

Nine individuals were recruited from prior researcher contacts. Additional participants were recruited from among vendors at the RSA conference [14], a large industry IT security conference that also hosts an exhibition floor with security-focused vendors. A list of 54 potential organizations was compiled after in-person researcher contact on the exhibition floor. After the conference, we identified organizations that provided organizational diversity in our sample and that were accessible to the researchers. We emailed 17 of them to invite participation in the study. Eleven organizations agreed to participate. One additional organization was recruited based on the recommendation of a participant.

## 3.2 Interviews

We collected data via semi-structured interviews. Interviews were conducted by two of the researchers and ranged from 30 to 64 minutes, lasting on average 44 minutes. We conducted 21 interviews with 1-3 participants per interview, 29 participants total. Five organizations opted to have more than one participant in the interview: three organizations had three participants and two organizations had two participants. Face-to-face interviews were conducted if feasible. Otherwise, participants were given the choice of a phone or video conference interview. Five interviews were conducted face-to-face, 10 by phone, and six via video. Interviews were audio recorded and transcribed by a third-party transcription service.

After the first nine interviews, we performed a preliminary analysis and chose to make minor revisions to the interview protocol in accordance with the qualitative research practice of theoretical sampling. Theoretical sampling involves adjusting data collection while the study is in-progress to better explore themes as they arise [13].

The interviews began with demographic questions about the organization (e.g., size, products) and the individual participants (e.g., role within the organization, professional background). Subsequently, participants were asked to describe their organizations' development and testing practices and associated challenges for their cryptographic products. Questions then transitioned into exploring cryptographic resources used by the organizations and how the participants thought those resources might be improved, if at all. The complete interview protocol is included in Appendix A.

## 3.3 Analysis

We utilized both deductive and inductive coding practices. Initially we constructed an *a priori* code list based on our research questions and literature in the field to provide direction in the analysis. As we performed multiple rounds of coding, we also identified emergent codes in the data. This iterative, recursive process helped us identify additional codes and categories as we worked with the data until we reached saturation [26, 69].

Five interviews (almost 24%) were first coded individually, then discussed as a group to develop a codebook. Although there is debate on the amount of text to collectively sample in qualitative research, the amount of text we group coded far exceeds the minimum of 10% often cited as standard practice [33]. We calculated intercoder reliability on this subset of the data using the ReCal3 software as a tool to help us refine our codes [21, 22]. For the five interviews, we reached an average Krippendorf's Alpha score of .70, with a high of .78, which is considered within the fair to good bounds for exploratory research having rich data with many codes and a larger number of coders [11, 15, 39].

Beyond the agreement metric, and in line with the views of many qualitative research methodologists, we thought it was important to focus on how and why disagreements in coding arose and the insights gained from discussions about these [5, 63]. These discussions better allow researchers to refine coding frames and pursue alternative interpretations of the data. During analysis, we found that each coder brought a unique perspective that contributed to a more complete picture of the data. For example, two of our coders more often identified high-level, nuanced codes about emotions and personal values, which may be due to their many years of working in human-focused contexts. These interpretations were often missed by the other coders who had more of a technology-focused background.

After coding of the initial subset of data, the remaining 16 interviews were coded by two coders each. Once each pair completed their coding, they had a discussion about the data to address areas of divergence about their use and application of the codes. This discussion resulted in the coders being able to understand each other's perspectives and come to a final coding determination. New codes that were identified during these discussions were added to the codebook, with previously coded interviews then re-examined to account for additions. The final codebook is included in Appendix B.

During the coding phase, we also engaged in writing analytic memos to capture thoughts about emerging themes [13]. For example, one memo captured thoughts on cryptography complexity. Once coding was complete, we reorganized and reassembled the data, created coding arrays, discussed patterns and categories, drew models, discussed relationships in codes and data, and began to move from codes to themes [59]. The team met regularly to discuss our emergent ideas and refine our interpretations. This process allowed for the abstraction of ideas and the development of overarching themes, such as how an organization's maturity and security culture drive formal development practices.

## 3.4 Limitations

Our study has several limitations. First, interviews are subject to self-report bias in which participants tend to underreport behaviors they think may be viewed as less desirable by the researchers, and over-report behaviors deemed to be desirable [16]. Given that the researchers who conducted the interviews represented an institution known for its security expertise, this bias may have influenced participant responses. We also note that the answers to some of the interview questions reflected participants' *perceptions* of the security level of their products and the security mindset of their organizations, which may or may not reflect reality.

Since there is no prior research into what is representative of the cryptographic development community, our sample is not characteristic of all types of organizations in this space. Although we did strive for diversity in organization size, with a smaller sample size common in qualitative research, we cannot definitively identify differences due to this variable.

# 4. DEMOGRAPHICS

Table 1 provides an overview of the organizations and participants in our study. To protect confidentiality, product types and participant roles are generalized.

The organizations represented in our study were of different sizes, with six being very large (10,000 or more employees), six large (10,00 - 99,99 employees), three medium (100 - 999 employees), three small (10 - 99 employees), and three very small/micro (1 - 9 employees) [24, 32]. All organizations developed a security product that uses cryptography (e.g. end user security software, hardware security module) or a non-security product that heavily relies upon cryptography to protect it (e.g. Internet of Things devices, storage devices, operating systems). Customers of these products ranged widely and included consumers, other parts of the organization, and organizations and businesses in multiple sectors such as government, technology, health, finance, automotive, and retail. Of the 15 organizations that discussed how long their companies had been implementing cryptography in their products, 12 had 10 or more years experience, with six of those having at least 20 years experience. The remaining three were startup companies that had been doing cryptographic development since their inception.

The 29 participants were a highly experienced group with several having made major contributions to the cryptography field. All participants had technical careers spanning 10 or more years, with several having been in the field for 30+ years. At least one individual from each of the interviews either currently worked on cryptography and security as a major component of their jobs (19 participants), or had

**Table 1: Interview Demographics**

| ID | Org Size | Reg | Prod Type | Participant(s) |
|---|---|---|---|---|
| C01 | VL | U.S. | HW | Lead crypto architect |
| C02 | VL | U.S. | COM | Lead cryptographer |
| C03 | VL | U.S. | HW, SW | Systems architect |
| C04 | VL | U.S. | HW | Crypto design reviewer |
| C05 | VL | U.S. | HW | Crypto architect |
| C06 | VS | U.S. | SW | Systems analyst |
| C07 | VS | U.S. | COM | Founder & researcher |
| C08 | VS | U.S. | IOT | Founder & developer |
| C09 | VL | U.S. | IOT | Researcher |
| C10 | L | U.S. | SW | Founder & engineering lead |
| C11 | L | U.S. | HW, SW | Product manager |
| C12 | S | U.S. | SW | 1) CTO<br>2) Marketing engineer<br>3) Business manager |
| C13 | S | U.S. | SW | 1) Chief Evangelist<br>2) Strategy Officer |
| C14 | M | U.S. | SW | 1) Marketing lead<br>2) Developer<br>3) Quality assurance |
| C15 | L | U.S. | SW | Principal engineer |
| C16 | L | U.S. | SW | CISO |
| C17 | M | Eur | SW | 1) CTO<br>2) Security engineer |
| C18 | L | Aus | SW | Crypto engineer |
| C19 | L | U.S. | SW | CTO |
| C20 | S | U.S. | COM | 1) Founder & architect<br>2) Compliance lead<br>3) Marketing director |
| C21 | M | Eur | SW | Crypto specialist |

**Org Size:** VL=Very Large, M=Medium, S=Small, VS=Very Small/Micro. **Reg** (Region/location of participant): U.S.=United States, Eur=Europe, Aus=Australia. **Prod (Product) Type:** HW=Hardware, SW=Software, COM=Communications Security, IOT=Internet of Things.

worked on cryptography extensively in the past (3 participants). The other participants were marketing or product leads, but all had a technical background.

Most of the participants had learned cryptography "on-the-job" as opposed to having formal training in the field. Five had an education in mathematics, but only two of those had studied cryptography as part of their formal study. Three had an engineering education, one had a physics degree, and the rest were educated in a computer-related discipline.

Four out of the 29 total interview participants had enhanced their knowledge through involvement in cryptographic standards groups. A cryptography architect commented on the value of his involvement in IEEE cryptographic standards early in his career: *"That's where I got to commune with cryptographers for a couple of years, me on the engineering side, and them on the crypto side... You end up learning things as a result of that process" (C05).*

## 5. RESULTS

The interviews revealed an organizational security mindset seldom seen in other cryptographic development studies. Our results suggest that the security mindsets had their roots in organizational attributes and culture that lay the foundation for the selection and use of cryptographic resources and rigorous development and testing practices.

For this section, we report counts of interviews throughout, joining group interview participants' answers to account for their organization. Due to the semi-structured nature of the interviews, the counts do not indicate quantitative results, but are reported to give weight to certain themes that were mentioned across interviews.

### 5.1 Security Mindset Characteristics

The interviews revealed organizational and personal characteristics that demonstrated a strong security mindset. These characteristics included professional maturity gained through experience, a deep understanding of the complexity of cryptography, and evidence of a strong security culture.

#### *5.1.1 Emphasis on Experience and Maturity*

Bruce Schneier said, "Only experience, and the intuition born of experience, can help the cryptographer design secure systems and find flaws in existing systems" [61]. The study participants expressed the importance of this experience as they repeatedly highlighted their own and their organizations' substantial maturity with respect to developing secure products and working with cryptography.

Overall, the organizations placed great value on hiring and retaining experienced technical staff. As previously mentioned, the majority of participants had substantial individual experience with cryptographic products. They also tended to work with other seasoned individuals. One participant described his team: *"We have a couple of the same core people on our test team who've been here for 25 years. They've gotten very good" (C01).* A startup company had only a few employees, but they all were veteran security software developers: *"Everyone we have has a lot of experience. I think the most junior person has a master's degree... and 10 and a half, 11 years of experience" (C07).*

Eight interviews noted the importance of experience when doing secure development, especially with cryptography. One participant remarked that secure products are ultimately dependent on *"the people that are designing it having the necessary knowledge and experience and understanding the whole picture, not just the little microscopic piece they're working on" (C01).* An interviewee with a long cryptographic background emphasized that there is no substitute for experience as he recounted a story of how his former company had to hire three less-qualified, full-time people to replace him in the work he had been doing on a part-time basis. A company founder remarked about the high level of technical maturity needed to properly deal with the complexity of cryptography: *"The level of education somebody needs to attain to be effective at doing crypto is relatively high. So it's not like I can put somebody who's fresh out of school on something and expect good results" (C10).*

#### *5.1.2 Recognition of Cryptography Complexity*

Based on their own experiences, participants and their organizations were keenly aware that, even though developing secure cryptographic implementations may appear to be easy, it is deceptively difficult. Despite proven algorithms being available, *"the algorithms are fairly involved,* *and they're difficult to understand" (C20).* Yet understanding the algorithm is just the first step. As described by an IoT researcher, translating the algorithm correctly into a product is *"not a trivial implementation" (C09).* One design reviewer remarked about the pervasiveness of cryptographic design errors he encountered over the course of his career: *"I think I reviewed about 2,500 products... I can only remember eight that did not have a problem that either... they had to fix, or... they had to change their marketing claims" (C04).*

Our interviews also suggest that building cryptographic systems appears to be more of an art than a science, requiring a careful balance between security, performance, and usability. One participant described these tensions:

> *"Crypto algorithms are already very highly optimized ... It's like balancing a supertanker on a 40,000-foot high razor blade, and if you make one small change you destroy the performance. If you make it the other way, you just destroy the security." (C04)*

Because of the complexity, our participants recognized that design and implementation errors can be rampant and require rigorous review and testing to answer the misleadingly simple question *"How do you know it's right?" (C08).* However, assessing cryptographic products can be challenging, requiring knowledge and experience to construct good tests. A cryptographic development architect described challenges his organization had in the past when they lacked maturity in cryptographic implementation and testing:

> *"We actually implemented a new symmetric encryption algorithm, and it passed all the tests... and it turned out that they did the algorithm completely wrong. That was because... they wrote a test which said, 'Generate some random data, encrypt it with the algorithm, decrypt it, and see if you get back the original data.' Well, yeah, it got back the original data, but the encrypted data was incorrect. It just was symmetric, so it did the same wrong thing encrypting and decrypting." (C01)*

The organizations also understood that cryptography is just one of many interdependent product components, with all of them having to be properly implemented to ensure security. This sentiment was echoed by one participant who commented, *"For us, the design of the overall architecture that uses the crypto algorithms is almost as important as the correctness of the underlying algorithms themselves" (C01).*

### 5.1.3 Security Culture

Each organization in our study appeared to have a strong security culture that was interdependent on the maturity and experience of its employees. A security culture is a subculture of an organization in which security becomes a natural aspect in the daily activities of every employee [60]. For development organizations, this involves having dedicated security people, spending money on security, making security a company core value, and offering secure products. For the studied organizations, the culture included a commitment to address security and the perpetuation of a security mindset to others in the organization.

**Commitment to Security:** The organizations we studied thought that having good product security and strong cryptographic implementations was a *"core value" (C07)*, *"the key to quality" (C09)*, and essential to company identity. As an example, a Chief Information Security Officer (CISO) of a large company remarked, *"In our company, we are developing and selling security to our customers. So we care about, basically, all three sides of the sort of security triangle [confidentiality, integrity, availability] in what we do." (C15).* A security engineer talked about his company's belief that security must be an important consideration even when faced with competing tensions such as time-to-market: *"Since we do a [security product], everybody feels that we need to add security and good crypto at every step, so it's not a big issue to find the right balance" (C17).* Another participant commented on how his company demonstrates its commitment to secure cryptography: *"We have some fairly large teams which concern themselves with cryptography and secure design methodology. All engineers get training on secure design and we make it a big deal in the company" (C05).*

Security culture is not just internally-motivated. External motivators, like gaining a larger market share or customer requirements, often necessitate strong attention to security. One participant commented on how customer expectations fostered a security culture that drove rigorous testing processes within his company:

> *"We serve the kinds of customers who rely on the stuff to work reliably and properly from the get-go, when they buy it. So it's not like... 'Maybe we'll update something later, if we find some problems.' That's not our philosophy, and that's not what our kind of customers expect... Part of that is also company culture." (C01)*

Although security culture is often thought of as a "top-down" phenomenon, it must be accepted by and acted upon at all levels. One participant, a CISO for a large company, commented on the importance of the security culture being pervasive throughout an organization:

> *"If there's senior executive support for a strong security program,... that helps tremendously. At the same time, if there is still a very strong feeling amongst a large number of developers that security, cryptography, and everything that's related to that is really a nuisance that should get out of the way and just to prevent them from writing more interesting features, it's definitely a concern." (C16)*

The interviews showed evidence that our participants are critical to the "bottom-up" support of organizational security culture. They serve as security champions and self-appointed educators, leading by example and projecting their values, personal philosophies, and commitment to security on the rest of the organization. Two of the participants explicitly embraced this role as a personal mission when they referred to themselves as "security evangelists." Another expressed his feeling of personal accountability to enact security in the products he supported: *"It's essentially a mark of my success or not, that I'm measured against, of whether those things remain secure or hacked" (C05).*

**Perpetuating a Security Mindset:** Just as the employees influence security culture, so does the culture influence employees by perpetuating a security mindset. Part of this perpetuation involves expert employees mentoring and supporting less-experienced personnel in their learning of secure programming methods and specialized security topics such as cryptography, as discussed in ten interviews.

The interviews suggest that providing an opportunity for individuals to gain hands-on experience with real products is important in understanding the issues involved in developing a cryptographic product. However, given the distinct possibility that a novice will make errors in the implementation, precautions must be taken. Two participants suggested that mock training exercises conducted on a separate testing infrastructure may be valuable initial steps. Others discussed mentoring and peer review activities within their organizations. For example, one organization enacted *"parent programming" (C14)* for any code that uses cryptography. Another had a formal peer review process:

> *"We review everyone's code... When I write something down and it has a flaw in it, I'm told about it, which is good... I think what we do is we take smart people who care about doing good work, and we foster an environment where they're not afraid to receive constructive criticism, and they're not intimidated away from giving it." (C15)*

The influence of an organization's security mindset does not necessarily end when an individual leaves that organization. As evidenced by three participants who left large companies to start their own small businesses, there seemed to be a transfer of security culture and practices from previous employers. A small company founder described this transfer: *"I think some of it could be just kind of from my career background, what I learned were the best practices... I think we've just kind of learned them in the beginning and kind of kept that culture" (C08).*

**Size Doesn't Matter:** We found no significant differences in perceived security culture or overall security practices based on size. Obviously, larger organizations had more resources to dedicate to security and cryptographic development. However, smaller organizations understood that vulnerabilities in their products could do great harm to the company's reputation, and so were committed to security and made thoughtful decisions about how they developed and tested, even if on a smaller scale. For example, the founder of a micro business noted:

> *"Being a small company, we're trying to also gain credibility. And we don't want to just claim that we have the fastest [crypto implementation] in the world, we also want to make sure that it is built safely and validated... [W]e cannot afford for this thing not to work properly." (C07)*

## 5.2 Selection of Resources

Because of security mindsets, participants revealed a proclivity towards careful selection of resources to help them attain their goal of secure cryptographic implementations. In this section, we describe considerations taken when choosing and evaluating cryptographic resources.

### *5.2.1 Standards*

In line with the popular quote "The nice thing about standards is that you have so many to choose from" by Andrew S. Tanenbaum [67], the interviews revealed that all the organizations used some type of cryptographic standards from organizations such as IEEE, ISO, American National Standards Institute (ANSI) [3], Internet Engineering Task Force (IETF) [37], and Payment Card Industry (PCI) [57]. All but one described using NIST standards or guidance documents, most commonly FIPS 140-2. The participants and their organizations were knowledgeable enough to understand and evaluate the appropriateness of cryptographic standards. However, not all organizations have the need to directly read the standards. For those that do, this requires maturity in the field given the standards' complexity. A Chief Technology Officer (CTO) at a small company expressed this challenge: *"A lot of standards are notoriously difficult to read. Unless you're an expert in the field, a lot of them don't make sense" (C12).* In another interview, a principal engineer commented on the difficulty in translating standards to products:

> *"I can tell you from my personal experience understanding the fundamentals of these things, still the standards were a challenge to use because they were very divorced from the implementation day-to-day details that I encounter while I'm trying to plug all the pieces together." (C15)*

Despite the complexity, when selected carefully and implemented correctly, standards were seen as beneficial for several reasons noted by our participants. Participants in eight out of 21 interviews commented that the community review of standards results in a more correct and secure solution. A CISO at a large software as a service company reflected on the value of public scrutiny: *"By relying on other standards that [were] vetted by multiple parties, I have much higher assurance that the underlying cryptography and design [are] done in an appropriate way" (C16).* A director of product management concurred with this: *"So the standards, because it's out there and everybody's looking at it and testing it, we depend on that as kind of a layer of security" (C14).*

Included in community review is the transparency of the standards process. One participant illustrated this observation using the popular AES standard as an example:

> *"It gave us a lot of comfort knowing how AES evolved ... being able to see all the steps, having that all happen out in the open and why and how it happened. Very helpful to us making the decision for what we're going to use and why we're going to use it." (C10)*

Participants in nine out of 21 interviews commented that the use of standards eliminates some of the difficulty of cryptographic development and testing by providing an authoritative foundation. The founder of a software company said his organization relies heavily on standards because *"inventing it on your own is just a different level of complexity that we knew enough to know we did not want to be involved in" (C10).* Standards also add confidence that a product will be *"interoperable with our customers, with our partners, even with our competitors" (C16).*

Finally, participants noted that standards-based products may elicit customer confidence. The director of product line management at a large credential management company spoke of the importance of customer trust in gaining market share, saying, *"If the standard is mature, then it means our product's going to be more easily accepted by customers" (C11).*

Standards meet the needs of most companies we interviewed; however, there are cases in which standards fail to address a specific need. In these situations, organizations may extend or modify standards. These extensions were viewed as adding rigor and security in addition to functionality, making the cryptographic implementations *"really ahead of any industry standard practices" (C05).*

Interestingly, three participants commented on distrust of government standards because of allegations that a U.S. government agency purposely weakened cryptographic algorithms [28]. For example, although one organization made extensive use of standards, they took special measures to exclude aspects of a government standard they felt were questionable and exercised extra rigor in their testing processes to account for potential vulnerabilities. In an extreme case, a consulting company's observation of customer distrust of government standards and their own frustration with the complexity of those led them to develop a new cryptographic primitive: *"I think it [the distrust] comes from . . . news reports or exposés that say this standard may not be as secure as we think. . . There is a lot of doubt out there . . . That's why there has to be additional options and alternatives" (C06).*

### 5.2.2 Certifications

Seventeen out of 21 organizations obtained at least one formal certification that the implementation of cryptographic algorithms in their products met standards specifications. Three additional organizations developed and tested to certification criteria without undergoing full certification. Eighteen organizations referenced FIPS 140-2 certification [49], five Common Criteria [41], three the Payment Card Industry Data Security Standard (PCI-DSS) [57] validation program, one the Underwriters Laboratories (UL) certification [70], and others pursued country-specific certifications.

The perception of the benefit provided by adhering to certification requirements was mixed among our participants. Among those who obtain certifications, only five organizations expressed that certifications establish additional confidence through independent testing. One of these remarked, *"You have a lot of assurance that everything's going to be tested and get that nice, kind of warm and fuzzy" (C08).* Six organizations noted that, even though they do not undergo the formal FIPS 140-2 certification process, they build to and test against the certification specifications to gain added assurance. A participant from a key and identity management software company stated, *"as a small company, I think it is actually extra important to make sure that we go through this battery of tests just to in a way reassure the people we're talking to that this is a robust product" (C12).*

For some participants, certifications are perceived as being more useful for meeting customer expectations than for bolstering security. Organizations most often obtain certifications because these are requirements of their customers in certain sectors (e.g. government, financial): *"for some areas, if you don't get the check-mark you don't get to play" (C11).*

Unfortunately, as noted in 12 of 21 interviews, certifications can be expensive in time and resources, making them prohibitive for smaller companies, especially those with products that run on multiple platforms and have frequent version updates. However, our interviews suggest that confidence in the cryptographic implementations may not be dependent on any certification, but rather on the rigorous development and testing practices these organizations undertake. The founder of a small company commented that FIPS 140-2 certification was too costly for them to pursue, but was confident that his product met the certification requirements: *"It's a place where we've done enough testing ourselves. I know it's fine. I know we would pass" (C10).* Another participant, whose company did undergo certifications, felt that the certifications provided little assurance beyond what was provided by their own internal processes:

> *"We always design and test ourselves to have confidence that [the product] meets all those requirements before we release it to the lab for their testing. So when they come back and say, 'It passed this,' we say, 'Well, okay, we expected that. Thank you.' So the surprise is if something fails, that we expected to pass. That doesn't happen very often." (C01)*

Four of our participants also remarked that certifications may not be a robust contributor to the security of a product. They expressed strong sentiments that certifications, especially FIPS 140-2, are more of a "checkbox" for customers *"without any additional benefit of security" (C02).* A CTO and long-time cryptographer added, *"FIPS 140 is. . . not focused on how to use crypto securely. It's focused on how to safely provide crypto functionality" (C19).*

In addition, seven out of 21 organizations commented that maintaining FIPS certification may, in some cases, weaken security by discouraging updates throughout the lifecycle of the product. Once a product undergoes a significant update (for example, fixing a security vulnerability), it may lose its certification. Organizations are then put in a difficult position: *"this ability to address vulnerabilities and to patch validated code is a real problem. It sends the wrong message if you do what you should do, which is patch it and live without [the certification]" (C19).*

### 5.2.3 Third-Party Implementations

The complexity of implementing algorithms from scratch and the expertise required to write those compelled two thirds of the organizations to follow industry best practices by not writing their own cryptographic code and instead using third-party cryptographic implementations, for example open source libraries such as OpenSSL or built-in operating system APIs. One participant used an analogy to explain why his organization used these resources: *"Not every person should be performing brain surgery on another person. I also don't believe every software engineer should really go write crypto code" (C07).*

The organizations selected these third-party implementations based on several factors. First, four mentioned an implicit trust of the resource based on the reputation of the vendor or general community acceptance. In describing why his organization chose a set of cryptographic libraries, one participant said, *"You pretty much trust those libraries because they are widely used, and you can run test vectors against them easily" (C20).* A third of the organizations said that they have more confidence in formally vetted, certified implementations. A participant from a small company that works on IoT cryptography commented, *"If a vendor has submitted a library through FIPS 140-2 certification, versus a code that was up on GitHub for example,. . . I would* *be more inclined to trust the one that has gone through the FIPS validation" (C08).*

Despite benefits of using third-party implementations, the organizations respected that the use of these external resources can still be difficult for less-knowledgeable individuals. A developer provided an example:

> *"[Crypto libraries] in general don't provide enough to be able to use them correctly out of the box... But there's many out there that think that they can just use AES, and I included it and I'm using it. But I'm not using it correctly, and then I'm leaving myself open to attack." (C14)*

This point illustrates that there is an important distinction between a cryptographic algorithm being certified or deemed "secure" and the proper use of the algorithm by developers. Similarly, third-party libraries have faced their share of security vulnerabilities due to implementation errors of otherwise sound cryptographic algorithms. Some of these vulnerabilities have had far-reaching impacts, for example the Heartbleed vulnerability in OpenSSL [71]. A security architect commented that third-party implementations are *"much more likely to contain implementation bugs and vulnerabilities" (C20).* Therefore, some organizations attempted to vet third-party implementations by doing their own vulnerability checks. One participant noted that his organization monitors the vulnerability databases for security issues with the libraries they use. Another commented on the extra security checks his company performs: *"We validate outside third-party libraries and software as they come in and confirm that they are bug-free and up to the latest standard or perform the right risk assessments" (C16).*

Finally, organizations may augment third-party implementations with their own internally developed modifications to avoid potential errors or address gaps in the resources. For example, to prevent developers from making errors while using the Windows CryptoAPI, a vulnerability assessment software company had *"written libraries on top of that to present a prettier facade in front of it because it's a fairly difficult library to use the way it's delivered" (C15).*

### 5.2.4 *Academic and Research Resources*

Our interviews revealed differing views on the value of academic research resources. Participants in three out of 21 interviews said that they have referenced academic resources (e.g., attended academic conferences or read (attack) research papers) to either better understand cryptography or to keep up with advances in the field. However, other participants passionately voiced their lack of confidence in the relevance of cryptographic research to their organizations' real-world industry challenges. One participant commented, *"People out in academia are famous for claiming there are holes in this kind of stuff where they don't actually exist, because they don't configure things according to the recommendations" (C01).* A cryptography architect asserted that his company's testing methods for cryptographic implementations were more state-of-the-art than those described in the academic world: *"No, we don't reference academic papers. They're not where we are in understanding the test problem... So there's a six-year gap between the... methods that we developed being identified in academia" (C05).*

## 5.3 Development and Testing Rigor

Our interviews revealed that the organizations translated their commitment to security and their expertise into rigorous development and testing practices. Interestingly, when participants spoke about how they test the cryptographic components, they saw secure software development as the foundation to providing cryptographic functionality.

### 5.3.1 *Formal Processes*

Of our 21 interviews, 20 reported that their companies employed formal development and testing strategies (those that are structured and standardized within the company) to ensure that their products, including the cryptographic components, were secure, while one participant said that they contracted developers to do that for them.

The development and testing practices were often the result of an evolutionary process spanning many years, as noted by 10 interviews. A director of quality assurance remarked, *"Part of [the role of] our test lead is now to verify that we're at the appropriate levels from a security standpoint... so it's a big focus for us now, whereas in the past, it was kind of a side item" (C14).* A company founder described his organization's introduction of more robust techniques over time:

> *"And it was an evolution, honestly. It started to where we didn't have hardly anything and new tools came on the market, as well as we had more time to focus on it. That allowed us to kind of improve code incrementally over time." (C10)*

Twelve interviews revealed a strong security mindset when they described developing and testing their products' cryptographic components based on risk. They would carefully build threat models to protect against strong adversaries, perform penetration testing, and would monitor current vulnerabilities to ensure their products were secure against those. A cryptographic design reviewer commented on this risk-based approach: *"All we can do is try to build good adversary models and then try to determine if our systems can stand up to those adversaries" (C04).*

Another discussed how his company's practices ensured that security had been considered throughout development:

> *"We have architects that do security reviews, that do threat modeling. And it's not just about the crypto, but more in general, how do you use the product. Who gets to do what? What are the risks? How do we mitigate those risks?... And one of the items for the engineering gate release is making sure... we mitigated anything that needed to be mitigated." (C11)*

Generally, the organizations' development and testing processes adhered to the principles in Figure 1. First, security specifications, architectures, and designs would be developed and reviewed. These would then be programmed against. Internal or external code reviews would be subsequently be conducted. During testing, tests would be run using internally developed test vectors or those provided with cryptographic resources. Static analysis tools and testing tools were also generally used. This development and testing process would be iterated whenever functionality changed, and performed in an expedited fashion if updates were being made due to a discovered security vulnerability.

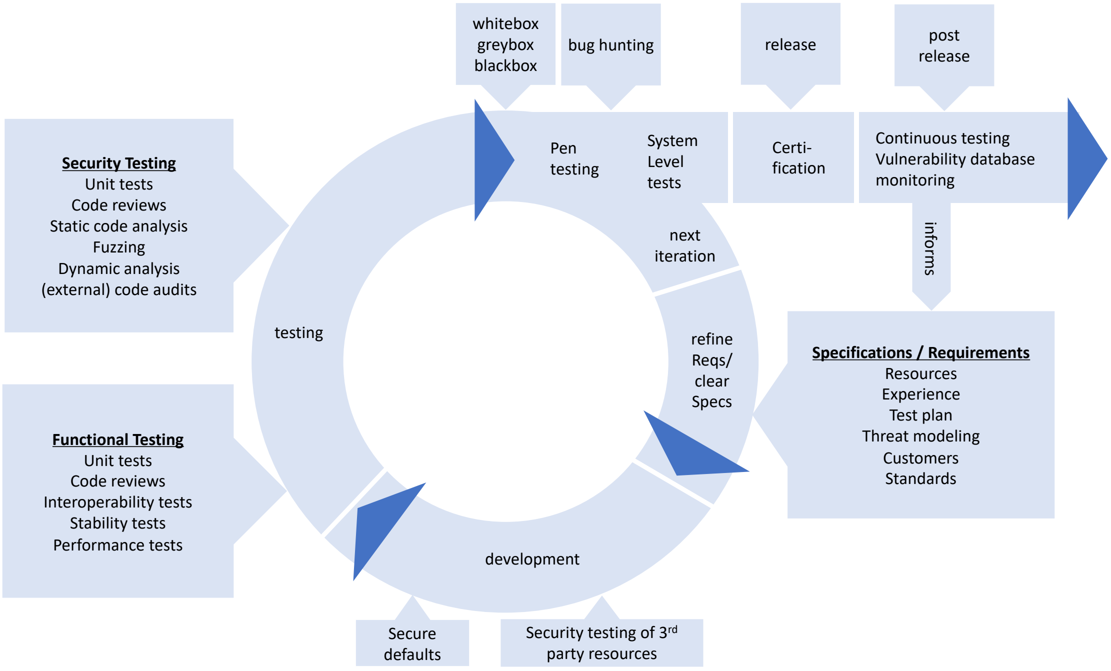

**Figure 1: Security Development Lifecycle [34] adapted to include development and testing strategies as extracted from the interviews. Not all processes apply to all interviewed organizations.**

### *5.3.2 Development*

As mentioned above, the development phase for the organizations followed typical, commonly accepted development practices such as requirements and risk analysis and programming. We specifically highlight two practices that were mentioned most often in the interviews.

Participants in nine interviews spoke about design reviews, which were critical for finding potential errors in cryptographic components early in the process. Those that do cryptographic review must be highly skilled and able to piece together others' thought processes:

> *"A lot of the review is really just archeology. It's delving down into what they're producing, trying to understand both the explicit and the implicit assumptions, and then identifying where there are conflicts that lead to attacks." (C04)*

Nine organizations also mentioned the importance of doing code reviews to look for security and functional errors and vulnerabilities. A participant from a security software company remarked, *"We have a mandatory and systematic code review. Each line of code and each comment of code needs to be reviewed by usually at least two peers" (C17).*

### *5.3.3 Testing*

Figure 2 shows the types of testing mentioned in the interviews. In 16 interviews, automated testing was discussed, often as being integrated with manual testing. The CTO of a company that produces security software discussed this integration of automated and manual testing: *"We have automatic tests, unit tests, integration tests, functional tests . . . , code analysis. . . . We have additional, manual tests being done. . . on top of the automated tests" (C17).*

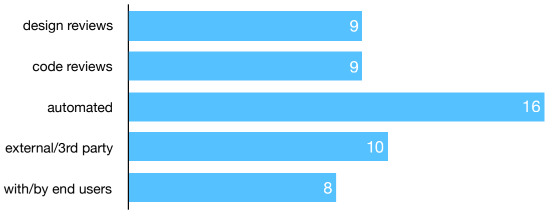

**Figure 2: Development and testing practices explicitly mentioned for secure cryptography.**

Ten interviews mentioned the use of third-party testing to improve the security of their cryptographic products. For example, organizations used bug-hunting services, or external testing such as blackbox, greybox, or whitebox penetration tests to increase the chances that bugs would be found in a controlled environment, and prior to product release. One participant described the benefit of his company using a bug-finding platform:

> *"You have some complete geniuses there that have found things that we never would have found. . . I feel like you're way more trustworthy if you are actually upfront about this stuff and you are actively soliciting people to attack you and paying them for their effort. You get a much higher confidence that some random person attacking won't just find something easy." (C10)*

Third-party review can also be used to gain trust with customers as expressed by a product manager: *"When customers ask us. . . 'Can you prove to me that it's done securely?' we can point to another organization that's independent to show" (C11).*

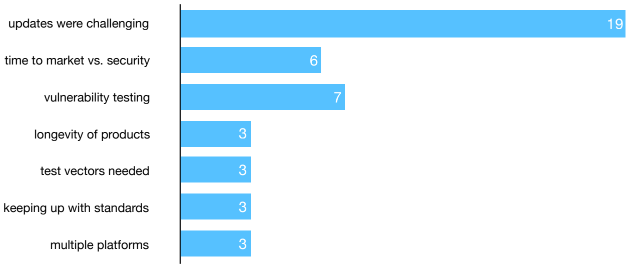

**Figure 3: Challenges in development and testing.**

End-user testing was not deemed as important for some organizations as they mostly develop products that become components in other products. However, this type of testing is critical for those producing products that will be directly used by businesses and consumers, and was discussed in eight interviews. These interviews mentioned formal beta-testing, continuous feedback, employing a user testing service, or recruiting convenience samples to test the product. One participant described the importance of user testing for his organization's consumer security software:

> *"We'd bring in our friends and family and sit them down and watch them. And it was eye-opening. It caused us to change a lot of what we did because, essentially, they didn't get the concept... We also used usertesting.com, which has been very great. You can show 10 people something and you have a pretty good idea." (C10)*

Another company takes advantage of beta testing to identify potential issues in their product: *"We have a very extensive beta program, and we have a very active customer base... so we have no lack of feedback" (C15).*

#### *5.3.4 Challenges*

All 21 of our interviews mentioned challenges to development and testing (Figure 3). We focus here on challenges directly related to cryptography, excluding challenges that have already been discussed, e.g., cryptographic complexity.

One challenge mentioned in six interviews was the tension between getting a product to market and taking the time to do robust security development and testing. For example, a participant from a company that spends years securely developing its cryptographic hardware modules observed that in most of the hardware/software industry, *"The design focus is simply not to do a solid job which will last a long time cryptographically.... They cannot care because they have to get the products out in a timely fashion" (C05).*

Testing for vulnerabilities in cryptographic implementations was another challenge mentioned in seven interviews, with three expressing concern for adequate testing of side-channel attacks. A participant who integrates cryptography into IoT devices said, *"especially in the embedded world, what the test vectors don't address I think is side-channel attacks, [which] could be really detrimental to the embedded device" (C08).*

Another challenge, as mentioned in three interviews, was the longevity of cryptographic products in customer spaces. An IoT researcher commented, *"Many devices are going to be deployed for 20 years. So, maybe the crypto in 20 years is no longer secure" (C09).* Another participant expressed the difficulty of having to maintain legacy cryptographic algorithms in a product: *"Old things become weak and you shouldn't use them anymore, and you need to add new ones ... However, customers are not so cooperative... It may take 10 years before everybody stops using something" (C01).*

Four interviews discussed challenges in having to troubleshoot or update third-party cryptographic implementations when vulnerabilities or errors are found. A principal engineer at a security software company described, *"when we're using any third-party library...and you happen to do something, and it fails, it's really hard to figure out what went wrong" (C15).*

Other cryptography-related challenges included the need for more test vectors (3 interviews), keeping up with changes in cryptographic standards (3), and having to use cryptographic libraries on multiple platforms (3).

In spite of these challenges, organizations in our study reported confidence in their processes and the resulting security of their cryptographic products (16 interviews). For example, a senior systems analyst at a small company described his confidence in the cryptographic algorithm they had developed: *"I don't make any bold claims, but at the same time, we looked at our encryption algorithm and we considered it quantum-proof" (C06).* In another interview, a company founder stated, *"I think we are definitely in the higher echelon for going above and beyond" (C10).*

# 6. IMPLICATIONS

## 6.1 Expanding Research Contexts

Our results suggested that the organizations believe they have a mature workforce, appreciate the complexity of crypto, possess a strong security culture, effectively use cryptographic resources, and practice secure development. However, the various studies mentioned in Related Work (e.g., [1, 2, 17–19, 42, 48]) indicate that there are many poor cryptographic implementations "in the wild," and developers typically lack a fundamental understanding of cryptography.

Where, then, is the disconnect between our findings and past research? It is possible that our self-report data are merely perceptions and do not accurately reflect the security mindsets of the organizations. Participants may be overconfident or may have overstated their organizations' security practices because of observer bias. We also recognize the value of future research to verify organizational claims by examining vulnerability databases, for example Common Vulnerabilities and Exposures (CVE) [68], to enumerate security issues in their products. Additionally, knowledge of an organization's development maturity level (e.g. Capability Maturity Model [12]) could be used as a comparison point.

Alternatively, this population is likely quite distinctive from previously studied developer populations and contexts, which may explain some of the differences in our findings. First, our participants exhibited more maturity in security and cryptography, with all having more than 10 years of security development experience. Second, organizational culture and constructs were an important driver of security mindsets within our study, while previous studies often involved independent application developers, many of whom were not full-time developers or had not received formal education or organizational training in programming, cryptography or security. Third, there was a marked difference in the types of cryptographic resources used. Other studies (e.g., [2]) indicated that developers are reliant on information gleaned from search engines and Stack Overflow, which was only mentioned once in our interviews. Instead, the organizations in our study turned to more authoritative resources such as cryptographic standards and certification specifications. Finally, many of the studies that identified cryptographic vulnerabilities examined mobile apps, presumably because these were easy to obtain from public application stores, and open source projects. Our participants were developing more complex, expensive software and hardware products. Lack of security in these products had greater consequences, with the potential of harming the company's reputation or resulting in loss of sales.

These differences suggest that perhaps the security research community is not capturing a comprehensive picture of the cryptographic development environment. This demonstrates the need for the research community to diversify their study populations and contexts, and consider mechanisms to bridge the gap between more security-mature and less-skilled developers who implement cryptography in their products.

## 6.2 Support for Other Populations

The evidence of the criticality of organizational security culture and collaboration during development and testing raises the question of whether it is even possible for "solo" developers, such as the application developers with little cryptography experience or peer support represented in previous studies, to be truly successful in this area. How then can the research community explore ways to facilitate the transfer of strong security practices observed in some organizations to others with less support and experience?

As previously stated, unlike the population in our study, many developers sampled in past studies rely on online communities such as Stack Overflow [65] when implementing cryptography. But there is little evidence that these communities provide the level of support necessary for secure development. Future research may involve further assessing the value of current online communities for cryptographic development and exploring alternatives as means by which security mindsets can be created and perpetuated.

The findings also reveal that mentoring and peer review are critical to perpetuating security mindsets within organizations. Past studies have sought to understand the effectiveness of software development mentoring, peer review, and pair programming (e.g., [7, 47, 74]). However, more work needs to be done to determine whether mentoring for cryptographic development requires different tactics and how to best support this outside of organizational constructs.

## 6.3 Cryptographic Resource Usability

Whereas the bulk of responsibility in producing secure cryptographic products lies with the organizations themselves, our results imply that cryptographic resource providers can also do more to contribute to developer confidence. The most mentioned complaint our participants had with standards and certification guidance was the complexity of the language. This underlies a need for standards organizations to work towards a common language between cryptography experts who write the standards and developers and engineers who use them. Although standards documents may not be the appropriate place for large amounts of explanatory text, supplementary guidance that contains more instruction, cautions against common errors, and provides example implementations may prove to be valuable. Just as research has been done on language for security alerts and warnings (e.g. [10, 66]), it would also be helpful for researchers to explore the efficacy of language that explains cryptography concepts to non-experts.

Additionally, developers could benefit from more explanations of motivation, in other words, the "why" behind cryptographic choices. One participant echoed this recommendation as an important way to move beyond the "checkbox" mentality of standards: *"So you're thinking big picture, 'I'm doing this for this a reason,' because otherwise, you just get in the cookbook approach of, 'Do I meet this? Yes, yes, yes. Check, check, check' " (C19).*

Of course, many developers have no need to look at the standards directly since they use third-party implementations. However, third-party implementations may also be difficult to interpret and use securely if one lacks basic knowledge of cryptography. Our study results support past research calling for increased usability of cryptographic APIs. Similar to the work of Montandon et al. that proposed a new platform for providing API code examples [46], we also recommend investigating new approaches to community vetting of sample code since developers often copy flawed code snippets from forums such as Stack Overflow [2].

Notably, the participants spoke of a security problem with FIPS 140-2: updating certified software for security would break its certification, so companies relying on the certification had to decide between withholding an update or having to undergo recertification. This insight into an instance where reliance on a certification can decrease security underlines the need to closely and continuously involve cryptographic experts in the certification process. Their experience as users of the certification can help evolve the process and shape it to be more resilient, usable, and secure.

# 7. CONCLUSION

Our study offers new insight into the cryptographic development and testing practices of a previously unstudied population of organizations and participants who were highly experienced in cryptographic development. Our results suggest that organizational security mindsets are based on maturity and a strong security culture, which in turn guide selection of cryptographic resources and inform rigorous development and testing practices. Based on these observations, we see opportunities for development organizations, cryptographic resource providers, and security researchers to facilitate an environment conducive to building the expertise required to correctly and securely implement cryptography.

## Disclaimer

Certain commercial companies or products are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the companies or products identified are necessarily the best available for the purpose.

## Acknowledgements

The authors would like to thank the following individuals who offered insightful comments that helped improve the quality of this paper: the anonymous reviewers, Curt Barker, Sascha Fahl, Simson Garfinkel, Michelle Mazurek, Matthew Smith, Brian Stanton, and Jeff Voas.

## 8. REFERENCES

[1] Y. Acar, M. Backes, S. Fahl, S. Garfinkel, D. Kim, M. Mazurek, and C. Stransky. Comparing the usability of cryptographic APIs. In *Proceedings of the 38th IEEE Symposium on Security and Privacy*, 2017.

[2] Y. Acar, M. Backes, S. Fahl, D. Kim, M. L. Mazurek, and C. Stransky. You get where you're looking for: The impact of information sources on code security. In *Proceedings of the 37th IEEE Symposium on Security and Privacy*, pages 289–305, May 2016.

[3] American National Standards Institute. ANSI. `https://www.ansi.org/`, 2018.

[4] S. Arzt, S. Nadi, K. Ali, E. Bodden, S. Erdweg, and M. Mezini. Towards secure integration of cryptographic software. In *Proceedings of the 2015 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward! '15)*, pages 1–13, 2015.

[5] R. S. Barbour. Checklists for improving rigour in qualitative research: a case of the tail wagging the dog? *British Medical Journal*, 322(7294):1115–1117, 2001.

[6] C. A. Barry, N. Britten, N. Barber, C. Bradley, and F. Stevenson. Using reflexivity to optimize teamwork in qualitative research. *Qualitative Health Research*, 9(1):26–44, 1999.

[7] A. Begel and B. Simon. Novice software developers, all over again. In *Proceedings of the Fourth International Workshop on Computing Education Research*, pages 3–14, 2008.

[8] D. J. Bernstein, T. Lange, and P. Schwabe. The security impact of a new cryptographic library. In *Proceedings of the International Conference on Cryptology and Information Security (LatinCrypt '12)*, pages 159–176, 2012.

[9] E. Bonver and M. Cohen. Developing and retaining a security testing mindset. *IEEE Security & Privacy*, 6(5), 2008.

[10] C. Bravo-Lillo, L. F. Cranor, J. Downs, and S. Komanduri. Bridging the gap in computer security warnings: A mental model approach. *IEEE Security & Privacy*, 9(2):18–26, 2011.

[11] H. Brenner and U. Kliebsch. Dependence of weighted kappa coefficients on the number of categories. *Epidemiology*, pages 199–202, Mar. 1996.

[12] CMMI Institute. What is capability maturity model integration (CMMI)? `http://cmmiinstitute.com/capability-maturity-model-integration`, 2018.

[13] J. Corbin and A. Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory.* Sage Publications, Thousand Oaks, CA, 4th edition, 2015.

[14] Dell Incorporated. RSA Conference: Where the world talks security. `https://www.rsaconference.com/`, 2018.

[15] K. DeSwert. Calculating inter-coder reliability in media content analysis using Krippendorff's Alpha. Technical report, Center for Politics and Communication, 2012.

[16] S. I. Donaldson and E. J. Grant-Vallone. Understanding self-report bias in organizational behavior research. *Journal of Business and Psychology*, 17(2):245–260, 2002.

[17] T. Duong and J. Rizzo. Cryptography in the web: The case of cryptographic design flaws in ASP.NET. In *Proceedings of the 31st IEEE Symposium on Security and Privacy*, pages 481–489, May 2011.

[18] M. Egele, D. Brumley, Y. Fratantonio, and C. Kruegel. An empirical study of cryptographic misuse in Android applications. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS '13)*, pages 73–84, 2013.

[19] S. Fahl, M. Harbach, T. Muders, L. Baumgartner, B. Freisleben, and M. Smith. Why Eve and Mallory love Android: An analysis of Android SSL (in)security. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS '12)*, pages 50–61, 2012.

[20] C. Forler, S. Lucks, and J. Wenzel. Designing the API for a cryptographic library: A misuse-resistant application programming interface. In *Proceedings of the 17th Ada-Europe International Conference on Reliable Software Technologies (Ada-Europe '12)*, pages 75–88, 2012.

[21] D. Freelon. Recal3: Reliability for 3+ coders. `http://dfreelon.org/utils/recalfront/recal3/`.

[22] D. G. Freelon. ReCal: Intercoder reliability calculation as a web service. *International Journal of Internet Science*, 5(1):20–33, 2010.

[23] R. Garcia, J. Thorpe, and M. Martin. Crypto-Assistant: Towards facilitating developer's encryption of sensitive data. In *Proceedings of the 12th Annual International Conference on Privacy, Security and Trust (PST '14)*, pages 342–346, 2014.

[24] Gartner. IT glossary: Small and midsize business (SMB). `http://www.gartner.com/it-glossary/smbs-small-and-midsize-businesses/`, 2017.

[25] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov. The most dangerous code in the world: validating SSL certificates in non-browser software. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pages 38–49, Oct. 2012.

[26] B. G. Glaser and A. L. Strauss. *The Discovery of Grounded theory: Strategies for Qualitative Research.* Transaction Publishers, 2009.

[27] M. Green and M. Smith. Developers are not the enemy! The need for usable security APIs. *IEEE Security & Privacy*, 14(5):40–46, Sept. 2016.

[28] L. Greenemeier. NSA efforts to evade encryption technology damaged U.S. cryptography standard. `https://www.scientificamerican.com/article/nsa-nist-encryption-scandal/`, Sept. 2013.

[29] G. Guest, A. Bunce, and L. Johnson. How many

interviews are enough? An experiment with data saturation and variability. *Field Methods*, 18(1):59–82, 2006.

[30] P. Gutmann. Lessons learned in implementing and deploying crypto software. In *Proceedings of the 2002 Usenix Security Symposium*, pages 315–325, 2002.

[31] J. M. Haney, S. L. Garfinkel, and M. F. Theofanos. Organizational practices in cryptographic development and testing. In *Proceedings of the 5th IEEE Conference on Communications and Network Security (CNS '17)*, Oct. 2017.

[32] B. Headd. The role of microbusinesses in the economy. `https://www.sba.gov/sites/default/files/`, Feb. 2015.

[33] R. Hodson. *Analyzing Documentary Accounts (No. 128)*. Sage Publications, 1999.

[34] M. Howard and S. Lipner. *The security development lifecycle Vol. 8*. Microsoft Press, Redmond, WA, 2006.

[35] Institute of Electrical and Electronics Engineers. IEEE Standards Association. `http://standards.ieee.org/`, 2018.

[36] International Organization for Standardization. Standards. `https://www.iso.org/standards.html`, 2018.

[37] Internet Engineering Task Force. IETF. `https://www.ietf.org/`, 2018.

[38] S. L. Kanniah and M. N. Mahrin. A review on factors influencing implementation of secure software development practices. *World Academy of Science, Engineering and Technology, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, 10(8):3022, 2016.

[39] K. Krippendorff. *Content Analysis: An Introduction to its Methodology*. Sage, 2004.

[40] D. Lazar, H. Chen, X. Wang, and N. Zeldovich. Why does cryptographic software fail?: A case study and open problems. In *Proceedings of the 5th Asia-Pacific Workshop on Systems (APSys '14)*, pages 7:1–7:7, 2014.

[41] D. Leaman. National Institute of Standards and Technology: NVLAP Common Criteria testing. NISTHB 150-20. `https://www.nist.gov/sites/default/files/documents/nvlap/NIST-HB-150-20-2014.pdf`, 2014.

[42] Y. Li, Y. Zhang, J. Li, and D. Gu. iCryptoTracer: Dynamic analysis on misuse of cryptography functions in iOS applications. In *Proceedings of the International Conference on Network and System Security*, pages 349–362, Oct. 2014.

[43] G. McGraw. Software security. *IEEE Security & Privacy*, 2(2):80–83, 2004.

[44] C. G. Menk. System security engineering capability maturity model and evaluations: Partners within the assurance framework. `https://csrc.nist.gov/csrc/media/publications/conference-paper/1996/10/22/proceedings-of-the-19th-nissc-1996/documents/paper010/cmmtpep.pdf`, 1996.

[45] S. Merriam and E. Tisdell. *Qualitative Research: A Guide to Design and Implementation*. John Wiley & Sons, San Francisco, CA, 4 edition, 2016.

[46] J. E. Montandon, H. Borges, D. Felix, and M. T. Valente. Documenting APIs with examples: Lessons learned with the APIMiner platform. In *Proceedings of the 20th IEEE Working Conference on Reverse Engineering (WCRE)*, pages 401–408, 2013.

[47] M. M. Müller. Two controlled experiments concerning the comparison of pair programming to peer review. *Journal of Systems and Software*, 78(2):166–179, 2005.

[48] S. Nadi, S. Krüger, M. Mezini, and E. Bodden. Jumping through hoops: Why do Java developers struggle with cryptography APIs? In *Proceedings of the 38th International Conference on Software Engineering (ICSE '16)*, pages 935–946, 2016.

[49] National Institute of Standards and Technology. FIPS Pub 140-2: Security requirements for cryptographic modules. `http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf`, 2001.

[50] National Institute of Standards and Technology. Cryptographic module validation program. `https://csrc.nist.gov/projects/cryptographic-module-validation-program`, 2016.

[51] National Institute of Standards and Technology. Cryptographic algorithm validation program (CAVP). `="http://csrc.nist.gov/groups/STM/cavp/`, 2017.

[52] National Institute of Standards and Technology. Cryptographic standards and guidelines. `https://csrc.nist.gov/Projects/Cryptographic-Standards-and-Guidelines`, 2018.

[53] P. Nguyen. Can we trust cryptographic software? Cryptographic flaws in GNU privacy guard v1. 2.3. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pages 555–570, 2004.

[54] Open Web Application Security Project. OWASP secure software development lifecycle project. `https://www.owasp.org`, 2017.

[55] OpenSSL Software Foundation. OpenSSL cryptography and SSL/TLS toolkit. `https://www.openssl.org/`, 2018.

[56] M. Q. Patton. *Qualitative research*. John Wiley & Sons, San Francisco, CA, 2005.

[57] PCI Security Standards Council. PCI Security. `https://www.pcisecuritystandards.org/pci_security/`, 2018.

[58] B. Potter and G. McGraw. Software security testing. *IEEE Security & Privacy*, 2(5):81–85, 2004.

[59] J. Saldaña. *The Coding Manual for Qualitative Researchers*. Sage, 3 edition, 2015.

[60] T. Schlienger and S. Teufel. Information security culture-From analysis to change. *South African Computer Journal*, (31):46–52, 2003.

[61] B. Schneier. Why cryptography is harder than it looks. `https://www.schneier.com/essays/archives/1997/01/why_cryptography_is.html`, 1997.

[62] B. Schneier. The security mindset. `https://www.schneier.com/blog/archives/2008/03/`, 2008.

[63] D. Seltzer-Kelly, S. J. Westwood, and D. M. Peña-Guzman. A methodological self-study of quantitizing: Negotiating meaning and revealing multiplicity. *Journal of Mixed Methods Research*, 6(4):258–274, 2012.

[64] S. Shuai, D. Guowei, G. Tao, Y. Tianchang, and

S. Chenjie. Modelling analysis and auto-detection of cryptographic misuse in Android applications. In *Proceedings of the 12th IEEE International Conference on Dependable, Autonomic, and Secure Computing*, pages 75–80, Aug 2014.

[65] Stack Exchange. Stack overflow. `https://stackoverflow.com/`, 2018.

[66] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: An empirical study of SSL warning effectiveness. In *USENIX Security Symposium*, pages 399–416, 2009.

[67] A. S. Tanenbaum. *Computer Networks: 2Nd Edition.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[68] The MITRE Corporation. Common vulnerabilities and exposures (CVE). `https://cve.mitre.org/`, 2018.

[69] D. R. Thomas. A general inductive approach for analyzing qualitative evaluation data. *American Journal of Evaluation*, 27(2):237–246, June 2006.

[70] Underwriters Laboratories (UL). Certification. `https://services.ul.com/categories/certification/`, 2018.

[71] US-CERT. OpenSSL Heartbleed vulnerability (CVE-2014-0160). `https://www.us-cert.gov/ncas/alerts/TA14-098A`, 2014.

[72] Veracode. The state of software security. `https://www.veracode.com/sites/default/files/Resources/Reports/state-of-software-security-volume-7-veracode-report.pdf`, 2016.

[73] Veracode. Veracode secure development survey: Developers respond to application security trends. `https://info.veracode.com/report-veracode-developer-survey.html`, 2016.

[74] L. Williams, R. R. Kessler, W. Cunningham, and R. Jeffries. Strengthening the case for pair programming. *IEEE Software*, 17(4):19–25, 2000.

[75] S. Xiao and E. Witschey, J.and Murphy-Hill. Social influences on secure development tool adoption: Why security tools spread. In *Proceedings of the 17th ACM conference on Computer supported Cooperative Work and Social Computing*, CSCW '14, pages 1095–1106, Feb. 2014.

[76] J. Xie and B. Lipford, H. R.and Chu. Why do programmers make security errors? In *Proceedings of the 2011 IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 161–164, Sept. 2011.

# APPENDIX

## A. INTERVIEW QUESTIONS

1. Can you tell me about your organization - what it does, what it produces?
2. What is your role within your organization with respect to cryptographic products?
3. How did you get into this field?
   (a) At what point and why did you become concerned with cryptography and secure development?
   (b) In which field(s) is your formal education?
4. Do you work in a unit or department that is part of a larger organization?
   If yes : What is the size of the unit or department?
   (a) What is the size of your overall organization?
5. Can you tell me about the kinds of products your organization develops, and specifically those that use cryptography?
6. Who are the typical customers for your products that use cryptography?
7. How long has your organization been working on products that use cryptography?
8. Is cryptography your organization's primary business focus, or is it an enabler within your products?
9. For your products that use cryptography, what processes or techniques , if any, does your organization use to minimize bugs and errors in code during the development process?
   (a) Why does your organization choose to use these methods? [only use if participant has difficulty coming up with response:] for example, industry standard, customer demand, robustness and quality
10. What processes or techniques does your organization use to test and validate the cryptography component in your products?
    (a) Why does your organization choose to use these methods? [only use if participant has difficulty coming up with response:] for example, industry standard, customer demand, robustness and quality
    (b) What kind of end-user testing, if any, does your organization do to prevent customers from misconfiguring or misusing the cryptography component in your products?
11. Does your organization do any certifications or third-party testing?
    (a) What reasons led you to decide to use certifications or third-party testing?
    (b) How do you establish confidence in the results of the certifications or third-party testing?
    (c) What are the challenges or issues your organization has experienced with certifications or third-party testing, if any?
12. What, if any, are your organization's biggest challenges with respect to developing and testing cryptography within your products?
    (a) How do you think these challenges can be overcome, if at all?
    (b) Has your organization experienced a tension between secure development and testing and getting a product to market? If so, how has that impacted your organization's processes?
13. Do your customers have specific requirements regarding development and testing? If so, what are those requirements?
14. How do updates impact your development and testing processes, if at all? (time-sensitive vs. deprecation)
15. What resources do you use to help you develop and test the cryptography component of your products? [only use if participant has difficulty coming up with response:] for example, standards, industry specifications, books, academic papers, standard libraries, APIs
    (a) What are the reasons your organization chooses to use those particular resources?
    If the participant does NOT use standards: What are the reasons that your organization does not use standards?
16. [If the participant uses standards:] What kinds of standards do you use?
    (a) What is the role of standards in your organization's development and testing processes?
    (b) What do you see as the value or benefit of using these standards, if any?
17. How could standards or other cryptographic resources be improved to be more useful?
    (a) How could NIST standards and guidance be improved to be more useful?
18. Is there anything else you'd like to add about the topics we've discussed?

## B. CODEBOOK

**Participant Demographics**

**Organization Demographics**

**Organization Characteristics**

- Team Interactions
- Security Culture
- Maturity
- Talent/Hiring

**Development and Testing Practices**

- Formal
- Informal
- Risk-based
- Practices
  - Automated
  - Human/Manual
  - External/3rd party testing
  - End-user testing
- Reasons/Philosophy
- Evolution
- Confidence
- Challenges
- Updates

**Certifications/Compliance Programs**

- Which ones (identify)
- Problems and Challenges
- Reasons
- Confidence
- Improvements

**Resources**

- Standards
- Government
- Industry/3rd Party
- Internally developed
- Research
- Gaps

**Security**

- Vulnerabilities and Errors
- Usability and Complexity
- Relationship and Tensions

**Security Education**

- Customers
- Developers/Engineers

**Emotions**

- Positive
- Negative

**Influences**

**Customers**

**Evolution of Security Field**

**Complaints**

**Participant Values and Perceptions**

**Trust**