Organizational Practices in Cryptographic Development and Testing

Julie M. Haney
National Institute of Standards and Technology
Gaithersburg, MD, USA
and
Department of Defense
Fort Meade, MD, USA
julie.haney@nist.gov

Simson L. Garfinkel and Mary F. Theofanos
National Institute of Standards and Technology
Gaithersburg, MD, USA
{simson.garfinkel, mary.theofanos}@nist.gov

***Abstract*—Organizations developing cryptographic products face significant challenges, including usability and human factors, that may result in decreased security, increased development time, and missed opportunities to use the technology to its fullest potential. To better identify these challenges, we explored cryptographic development and testing practices by conducting a web-based survey of 121 individuals representing organizations involved in the development of products that include cryptography. We found that participants used cryptography for a wide range of purposes, with most relying on generally accepted, standards-based implementations as guides. However, many also developed their own implementations and drew on non-standards based resources to inform their development and testing processes. Our results also highlight challenges that incorporating cryptography within products creates within organizations, including the recruitment and management of talent, the product lifecycle, and the ability to explain the security value of products to customers. We conclude by discussing implications of these findings and opportunities for future research.**

***Index Terms*—Cryptography, Usability, Cryptographic standards, Developers**

I. Introduction

As a community of security researchers and practitioners, we are increasingly aware of the pervasive impact that usability and human factors have on the security of information systems. This is especially true in the case of cryptographic development resources (e.g. APIs, standards, and tools) where usability is notoriously poor and has long been regarded as a barrier to development [1], [2]. Cryptographic testing resources fare no better. Certification and testing programs such as the National Institute of Standards and Technology (NIST) Cryptographic Algorithm Validation Program (CAVP) [3] and the National Information Assurance Partnership (NIAP) Common Criteria Protection Profiles [4], [5] currently fail to address usability concerns. Church et al. [6] made a similar observation in 2008, noting that certification procedures that do not consider usability frequently make "unrealistic assumptions as to what the user is capable of" and, as a result, produce a "false perception of what is being assured."

Before making recommendations for increased usability in cryptographic development and testing resources, we must first understand our target population: the organizations and developers that use these resources. Therefore, our research explores the practices and challenges of organizations that are developing products that use cryptography. Our hope is to use this new understanding to improve cryptographic tools and inform greater usability of cryptographic resources.

In this paper, we present the results of a survey of 121 individuals working in organizations that implement cryptography in their products. The survey was guided by the following research questions:

- RQ1: What sources and resources are used for cryptographic implementations?
- RQ2: What cryptographic test and evaluation approaches are employed?
- RQ3: What factors are important to organizations when evaluating the quality of a cryptographic implementation?
- RQ4: How are cryptographic standards used when developing products that implement cryptography?
- RQ5: What challenges do cryptographic developers and their organizations face that are greater than or different from those with non-cryptographic products?

Our research has several contributions. We believe we are the first to ask broader questions to characterize the cryptographic practices and types of resources and standards used by cryptographic developers within *organizations*. Our survey sample differs from many of the cryptography developer research studies to date in that our participant pool consists of professionals who represent organizations and work in cryptographic development primarily on a full-time basis vs. the students or part-time app developers in other studies, for example [7]. This research also offers insights into the

challenges that cryptographic implementations introduce into organizational practices from the conceptualization of the product; the assembling of the product team; the design, implementation and testing of the product; and finally to the marketing, sale and end-user support. We believe that this is also the first study to attempt to quantify and rank factors that organizations consider when evaluating the quality of a cryptographic implementation.

II. Related Work

We do not discuss the extensive literature regarding end-user usability of cryptographic products because our focus is on the usability issues of creating and testing those products, not their use.

Decades of research into cryptographic primitives and the development of cryptographic libraries has created a plethora of choices for developers wishing to integrate cryptography into their products. Many security vulnerabilities have resulted from the inability of developers to successfully navigate these choices and securely assemble the cryptographic components into an application. Gutmann [8] observed in 2002 that the powerful functionality of cryptographic libraries frequently resulted in the introduction of security bugs by software developers who did not understand the implications of their choices. In 2004, Nguyen [2] showed that even open-source implementations that are under public scrutiny have cryptographic flaws. Eight years later, Fahl et al. [9] analyzed 13,500 free apps downloaded from the Google Play Store and found that 8% were susceptible to man-in-the-middle attacks, including apps from major financial institutions and established internet providers. Using static analysis, Egele et al. [10] found that 88% of 11,748 examined applications contained a significant error in their use of a cryptographic Application Programmer Interface (API).

Acar et al. [7] performed a systematic analysis of how information resources available to Android app developers impacted code security. They found that participants who were only allowed to use Stack Overflow produced code that was significantly less secure—but more functional—than those who used only the official Android documentation. Nadi et al. [11] analyzed the top Stack Overflow posts on cryptography and, correlating with data from GitHub, concluded that cryptographic APIs were too complex to use reliably, required understanding of the underlying API's implementation, and were at the wrong abstraction level to allow developers to perform common cryptographic tasks.

So what is to be done? One approach is to help developers make better use of APIs and other resources by improving usability and guiding developers to make secure choices [1]. To assist developers, Artz et al. [12] created an interactive plugin for the Eclipse integrated development environment (IDE) to assist Java developers in choosing and integrating the appropriate cryptographic algorithms. Crypto-Assistant [13] and Crypto Misuse Analyzer (CMA) [14] performed similar tasks. However, no user evaluations of these systems were performed, so there remains the of question whether these tools actually result in developers writing code that is more secure. Acar et al. [15] explored the usability of several Python cryptographic APIs. They found that clear documentation with easy-to-use code samples and support for common cryptographic tasks (for example, secure key storage) may be just as, or more important than simple interfaces.

Others alternatively have proposed cryptographic APIs sitting atop new libraries. Forler et al. [16] developed libadacrypt, a cryptographic library written in Ada that is designed to be "misuse-resistant." Bernstein et al. [17] created the Networking and Cryptography library (NaCl), a cross-platform cryptographic library that is designed and implemented to "avoid various types of cryptographic disasters suffered by previous cryptographic libraries such as OpenSSL." Although both libadacrpyt and NaCl were designed to be easier for developers to use, to date, these libraries have not been formally tested for developer usability.

III. Methodology

Between June and July 2016, we conducted a twelve-question, web-based survey targeting individuals with experience developing products that include cryptography. Participants were asked questions related to their cryptographic implementation practices and the challenges their organizations face. The survey questions, informed by discussions with cryptography experts in our organization, contained closed-ended, multiple response ("check-all-that-apply") and five-point scale items, as well as open-ended, free-response items.

The study was approved by the NIST Human Subjects Protection Office. Survey responses were collected and recorded without personal or machine identifiers (e.g., IP address) and not linked back to the recruitment emails or participants. Each survey participant was assigned a reference code that was used for all associated data in the study.

We sent email survey invitations to over 800 individuals on government-industry partnership cryptography mailing lists, 68 individuals from 49 unique companies who had spoken at applied cryptographic development conferences, and 89 U.S. organizations from the Worldwide Encryption Products Survey [18]. Participants were not required to complete all survey questions, but only those who substantively answered at least five of the 10 non-demographic questions were considered. A total of 121 responses were included in the final study results.

For the open-ended responses, we performed iterative, inductive coding on the data to identify general themes. Inductive coding is a commonly used qualitative data analysis approach that allows findings to emerge from the text without the restraint of having to use pre-defined, structured categorizations [19], [20].

Table I
PARTICIPANT JOB FUNCTIONS

Job Function Category	n=	%[a]
Managerial (e.g. executive, program or department manager)	17	14%
Cryptographer	11	9%
Developer/Software Engineer	17	14%
Researcher/Educator	9	7%
Security Professional (e.g. security architect, security engineer)	10	8%
Technical - Executive (e.g. CTO, Chief Scientist, Technical Director)	12	10%
Technical - Other (e.g. architect, engineer, certifications)	21	17%
Unknown/not specified	24	20%

[a]Note: percentages do not sum to 100% due to rounding.

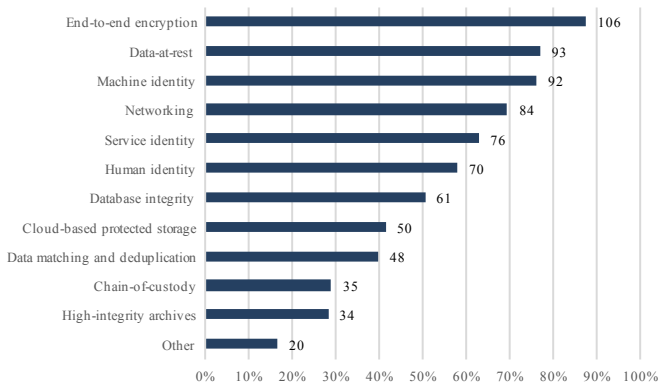

Figure 1. Use of cryptography in products, 769 mentions, 121 respondents.

IV. SURVEY FINDINGS

Because not all participants answered all questions, we will specify the number of responses for each question included in our results. For example, if 40 out of 119 participants responding to a question selected a particular option, we will indicate that with the notation of "40/119." Also, since there were several multiple responses questions, we also report the number of "mentions" for those questions, in other words, the number of total options that were selected by all respondents for that question. Some response options listed below are abridged from the actual survey text for readability purposes.

Specific participants are referred to with the notation P# where # is between 1 and 121, for example, P23. When providing direct quotes, the participant ID will be followed by a description of their job function (when available). Direct quotes from participants are provided only for those participants who granted permission to quote their responses.

A. Participant Job Functions

Ninety-seven participants specified at least one job function. We categorized reported job functions into high-level categories (see Table 1). Lack of specificity in these open-ended responses (e.g. "engineer") made it impossible to accurately categorize all functions, so a best estimation was made.

Our participant pool showed an obvious skew toward technical roles (80/121), which is appropriate for our research goals. Note that job function categorization into a managerial role may not necessarily mean that the participant lacks technical expertise. The job function responses in our survey also indicated that most of our participants worked on product development or had a role in an organization performing development as their primary job.

B. Product Descriptions

In order to provide some insight into the kind of products represented in the survey, participants were asked, "Where is cryptography used in your products?" Fig. 1 summarizes the responses, showing that end-to-end encryption, data-at-rest, and machine identity were the most commonly specified types of cryptography. Of participants who checked the "Other" option, the two most commonly mentioned categories were software/firmware code integrity (5/20) and hardware-level encryption and security (3/20).

We also asked participants to describe what their product does and how it uses cryptography. It was difficult to categorize these open-ended responses due to the variation in detail and terminology used by participants, so we were not able to do more in-depth statistical analysis of participant responses in relation to product type. Out of the 103 responses, the majority of products were software-based, with only 15 indicating a hardware-based solution. Products were diverse and included operating systems, cryptographic toolkits and libraries, internet of things devices, disk and file encryption, network communication encryption, certificate authorities, authentication software, and key-management solutions, among others.

We also attempted to categorize product descriptions into the number of cryptographic products that each participant represented in the survey. Of the 103 respondents, 40.5% of participants had a single product that uses cryptography, 26.45% had more than one product (but not many products), and 12.4% were from an organization with a large product line. In addition, 1.65% of products were research prototypes or proofs of concept.

C. Cryptographic Implementation Sources and Resources

We asked participants about the sources of the cryptographic implementations used in their products with 82/120 (68.33%) indicating that they use what the hardware, operating system, and standard libraries provide. Almost as many (80/120, 66.67%) said that they develop their own cryptographic implementations and 69/120 (57.5%) selected open-source implementations. Less common were the purchase of commercially available

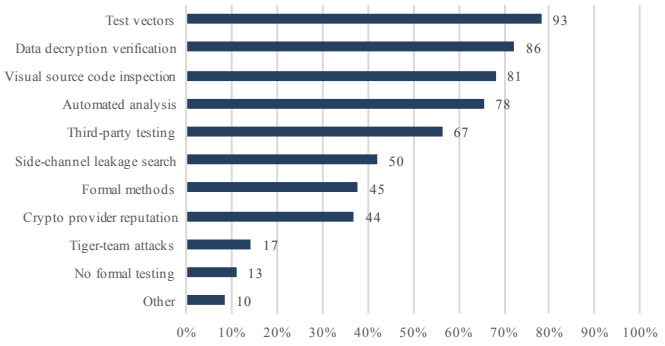

Figure 2. Test and evaluation approaches, 584 mentions, 119 respondents.

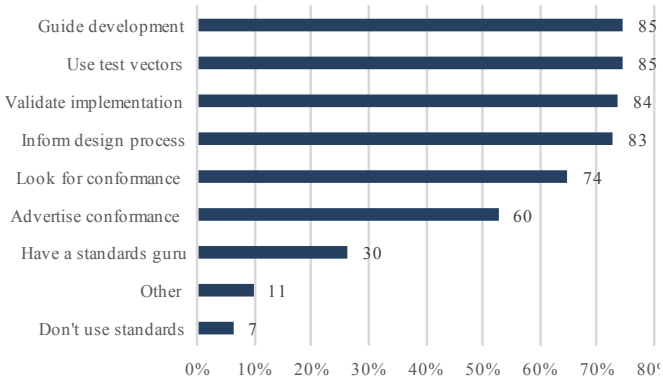

Figure 3. Uses of cryptographic standards, 519 mentions, 114 respondents.

implementations (37/120, 30.83%), requiring customers to purchase specific cryptographic implementations to use with their products (24/120, 20%), and contracting with others to develop proprietary implementations (15/120, 12.5%).

Participants were also asked what resources they use to help them select or develop cryptographic implementations. Out of 117 responses, an overwhelming majority said that they use national or international standards (109, 93.16%) and industry specifications (94, 80.34%). 66.67% (78) use academic literature, 44.44% (52) use internal (corporate) guidance, 37.61% (44) use web sites, 34.19% (40) use reference books, and 29.06% (34) use proprietary information.

D. Test and Evaluation Approaches

There are a wide variety of approaches that organizations may use for testing, evaluating, and gaining confidence in their cryptographic systems and implementations. To gain greater insight into the incidences of these approaches, we probed participants about their test and evaluation practices. Results are in Figure 2. Ninety-three out of 119 respondents (78.15%) indicated that they test their implementations with specific test vectors (test cases), while 86 (72.27%) said that they verify that the data can be decrypted after it is encrypted. Thirteen (10.92%) said that they do not do formal testing, but would be able to visually observe if data were not being properly encrypted.

Cryptographic certifications are often a customer purchase requirement for cryptographic products. For example, third-party testing laboratories may perform validation testing to the CAVP and NIAP guidelines mentioned earlier. More than half (67/119, 56.3%) specified that they rely on third-party testing for certification or aim for compliance with testing programs. In a related, open-ended question, participants were asked if they employ a third-party testing organization and which one(s) they use. Fifty-three respondents indicated some use of third-party testers, with 21 listing at least one commercial testing lab by name; 20 mentioning NIST, a NIST validation program (e.g. CAVP) or the Federal Information Processing Standard (FIPS) [21]; and 11 mentioning the Common Criteria or NIAP. As discussed in the Limitations section, the participant pool was weighted towards those with an awareness and interest in NIST-related cryptographic resources, which may explain the observed bias towards government-endorsed testing programs.

E. Use of Cryptographic Standards

The survey asked participants if they use cryptographic standards and, if so, how. Approximately 74% (85/114) use standards to guide their development process, with the same number using test vectors from the standards to validate the cryptographic modules. 6.14% (11/114) actually indicated that they don't use standards. See Figure 3 for the response summary.

In two open-ended response questions, participants were also asked which cryptographic standards they use and how they chose those standards. For the types of cryptographic standards, the level of detail provided by respondents varied greatly. Some participants listed specific algorithms (e.g., AES, SHA2, 3DES), others mentioned standards bodies (e.g., IEEE) or policies (FIPS, NIST SPs), and still others indicated the use of standards in generic terms, for example, *"Too many to list"* (P16).

As we were most interested in the sources of standards, we attempted to quantify these open-ended responses by doing frequency counts on mentions of standards authorities. Out of 83 responses, the six most frequently mentioned standards organizations were NIST (including mentions of FIPS) (61), IETF (20), ANSI (14), International Organization for Standards/International Electrotechnical Commission (ISO/IEC) (14), IEEE (13), and Common Criteria/NIAP (12).

Reasons for the choice of cryptographic standards varied. We categorized the 72 open-ended responses, revealing that

Table II
IMPORTANCE RANKING OF EVALUATION METRICS FOR CRYPTOGRAPHIC IMPLEMENTATIONS

Rank	Evaluation Metric	Mean	Score 4 or 5	Score 1 or 2
1	General acceptance of algorithms	4.20	81%	10%
2	Runtime protection of secrets	3.92	68%	13%
3	Documentation quality	3.89	66%	12%
4	Code quality	3.88	72%	11%
5	Support for HW that accelerates crypto operation	3.75	63%	15%
6	Throughput	3.73	66%	15%
7	Support for HW that supports crypto key stores	3.69	61%	19%
8	Evaluation by third-party testing organizations	3.68	62%	19%
9	Openness of source code	3.38	48%	27%
10	Implementation language	3.34	49%	25%
11	Implementation size	3.26	42%	27%
12	Evaluation by outside trusted consultants	3.24	46%	27%
13	Power requirements	3.20	43%	28%
14	Blessed by local expert	3.05	36%	35%
15	Popularity in other products	2.92	35%	36%

the most popular reason (31, 43.06%) was market drivers and customer demand, followed by government mandate and recommendations (19, 26.39%), applicability to the domain/problem (10, 13.89%), the quality of the standard (7, 9.72%), interoperability (7, 8.33%), prior familiarity or experience with the standard (5, 6.94%), and internal organizational assessment (4, 5.56%).

F. Quality Evaluation Factors

Participants were asked to individually assess the importance of 15 different metrics, or factors, when evaluating the quality of a cryptographic implementation, with a rating of one being the least important, and five being the most important. 120 participants ranked at least one of the factors. Table 2 shows the resulting ranked order.

There was a notable difference between job function responses. We include the top three and bottom three scored factors for each group in Table 3. General acceptance was ranked highly by all groups except Technical Executives. Popularity was ranked low by all groups. However, there was variation in other factors deemed to be important by each group.

G. Organizational Challenges

In a multi-part, open-ended question, participants were asked about the challenges their organizations face when incorporating cryptography in its products, specifically where cryptography creates challenges that are fundamentally greater or different than other kinds of technology.

1) Recruiting Talent: When asked about recruitment challenges, the majority (42/66, 63.64%) indicated that it was hard to find qualified talent with strong cryptography skills. Three participants commented on the challenge of evaluating cryptographic and secure coding expertise in an interview setting. Since cryptography is a niche skill, these findings are not surprising. However, recruiting employees with just cryptographic knowledge is not the only goal. Nine participants mentioned the need for employing talent who have a combination of skill sets. P22, a software engineering manager at a large company, said, *"It is hard to combine software engineering, math, cryptography, and collaboration skills into one person who would understand the importance of shipping a secure product."*

Table III
MOST AND LEAST IMPORTANT FACTORS PER JOB FUNCTION. () INDICATES A TIE.

Job Function	Top 3 Factors	Bottom 3 Factors
Managerial	Runtime Throughput Acceptance	Openness Popularity (Local expert, Power, Size)
Cryptographer	(Runtime, Third party eval.) (Acceptance, HW crypto key) (Throughput, Documentation)	Popularity Language Openness
Developer/ SW Engineer	Code quality Openness Acceptance	Power Size Popularity
Researcher Educator	Acceptance Runtime Openness	Expert eval. (Popularity, Outside eval.) Language
Security Professional	HW crypto key (Acceptance, Code quality, HW acceleration) (Documentation, Size)	(Popularity, Openness) Language Power
Technical Executive	HW acceleration HW crypto key (Documentation, Throughput)	Expert eval. (Popularity, Outside eval.) Power
Technical Other	Acceptance HW crypto key (Runtime, Documentation)	Popularity Openness Language

To address recruitment problems, eight respondents said they use a hire-and-train strategy. One participant wrote, *"We generally end up hiring smart people and training them on security and cryptography"* (P19, director of development at a security consulting company).

2) Managing Employees and Evaluating Employee Work: Only 17/43 of respondents (39.5%) indicated that there were greater challenges when managing employees. However, nine of these participants indicated that having highly intelligent employees with specialized skills introduces management challenges, especially when managers are not well-versed in cryptography. Furthermore, traditional project management must be adjusted to accommodate the meticulous nature of building quality cryptographic implementations and the characteristics of those working in that field as expressed by one participant: *"Cryptographers tend to drill deep and act conservatively so expectations around how quickly a task will get done need to be adjusted"*

(P106, engineering manager at a large software company).

When asked about challenges in evaluating employee work, a little more than half (22/40, 55%) indicated that they face greater challenges. Seven participants commented that an expert or expert source is needed to appropriately evaluate cryptographic work, which is a challenge given the dearth of cryptography experts. Three participants said that it is hard, and perhaps impossible, to know when employee work has reached a truly secure level.

3) Obtaining Appropriate Development Tools: When asked about obtaining appropriate developer tools, 13/40 (32.5%) of responding participants felt that there were greater or different challenges. Of those, the most commonly mentioned challenge (6/40) was in acquiring and having the expertise to use quality bug-finding and testing tools.

4) Maintaining and Transitioning Products: Responses about challenges in product lifecycle maintenance (45 responses) and transition to new products (43 responses) revealed several shared themes. The most common theme, mentioned by 17 participants, related to challenges with maintaining backwards compatibility with older cryptographic algorithms. One participant in particular clearly articulated challenges with the staying power of deprecated cryptography: *"Crypto protocol agility has created a massive problem of zombie algorithms that can't be got rid of. Transitioning to new algorithms is easy. Transitioning from old algorithms is hard"* (P32, random number generator expert at a large company).

Participants further noted a conflict between moving towards state-of-the-art cryptography and maintaining the functionality of legacy applications which are often employed for many years at a time. One participant remarked, *"Old crypto algorithms never die; we'll never get rid of them because somewhere, someone, many years ago, protected data with an old algorithm"* (P95, software engineering manager at a large software company). Another said, "*There is a tension between turning off weak and dangerous algorithms and not breaking users"* (P118, cryptographic library developer).

The update of cryptographic products due to security vulnerabilities or bugs was viewed as challenging by 16 participants. The frequency of security issues in foundational cryptographic implementations and libraries, in particular, requires extra effort as expressed by one participant: *"This is a huge problem. We are constantly needing to evaluate security vulnerability announcements, updating our own products, and then advising customers to update their systems"* (P66, CTO, cybersecurity vendor).

Because of the seriousness of vulnerabilities in cryptographic implementations, several participants commented that updates demand an urgency above and beyond that of other technologies and can be disruptive to the organization. One participant commented, *"Security patches are disruptive and propagate all the way up from the cryptography toolkits through to many of the company products"* (P106, engineering manager at a large software company).

Another theme emerging from the survey data of nine participants was a challenge in keeping abreast of changing standards. One respondent noted, *"Main issue is getting product teams to upgrade to address every changing FIPS requirement and crypto strength requirement"* (P52, software crypto module development engineer). This was an interesting perception given that, in actuality, FIPS standards change infrequently.

A final theme was customer resistance to transitioning to new products, noted by five participants. These participants commented that customers are often hesitant to adopt new cryptographic standards and products, even if they offer greater security. Said one respondent, *"Customers view crypto and crypto libraries as...unchanging components of their systems"* (P65, senior engineering manager at a large vendor).

5) Participating in Product Evaluations: Out of 31 responses, 20 participants indicated that they face greater or different challenges when participating in cryptographic product evaluations. No common themes emerged from these data. However, three participants did note challenges when bringing together multi-disciplinary evaluation teams with different foci and expertise. One participant said:

> *"As usual in the world of security, many product engineers aim at having something working, while cryptographers and security folks focus on the opposite: does it fail where it is supposed to? Secondly, explaining a theoretical cryptography game for an adaptive adversary to a standard software engineer or a program manager simply induces sleep and results in lack of credibility for the cryptographer...[T]he difference in disciplines is just too large."* (P95, software engineering manager)

6) Explaining Products to Potential Customers: Explaining products to customers was viewed as a challenge by 45/48 respondents. Seventeen of these participants said that customers lack security and cryptographic knowledge, making it more difficult to explain a product's functionality or for the customer to properly use the product. One participant commented, *"Many potential customers do not know anything about PKI and our products are all PKI-based so we often end up having to teach them PKI in order to explain our products"* (P19, director of development at a security consulting company). Furthermore, two participants noted that they don't believe that customers even care to understand the underpinnings of the product, with one bluntly stating, *"90% just want a certification check off"* (P22, certifications coordinator for a secure mobility products company).

Nine participants commented that it can be difficult to demonstrate the value proposition of their products, and four noted customer concerns about paying for increased security or more robust implementations when what they

already have seems to be "good enough." P98, a storage security engineering consultant, commented, *"security adds assurance (insurance) at a cost, but does not provide a way to improve profit."* Another participant wrote about competition with open source implementations: *"In the commercial sector, we have had difficulty convincing potential customers/partners on the value of our products vs. 'free' open source products even with data"* (P39, CEO of a small security solutions company).

Difficulties in convincing customers about the value of cryptographic products may stem from the observation of four participants that many customers view security as an impediment. One participant commented, *"Security is an inconvenience and complicated. Customers need to know a lot to use the products correctly and securely but they don't understand it easily or willingly"* (P106, engineering manager at a large software company).

Five participants remarked on challenges relating to the varying levels of detail they have to provide to their potential customers and what that appropriate level of detail is. Providing detail allows for greater exposure of the strengths of the product, but may also confound the product explanation when customers aren't especially knowledgeable about those details. For example, one participant commented:

> *"Customers don't need to know the details of the crypto unless they're experts, in which case they can understand. The only thing that causes problems is when non-experts insist on having things explained or seeing lab reports and then don't know how to interpret the answers."* (P30, CTO at a cryptographic platforms and services company)

A final theme voiced by four participants was the role of marketing. Three of these participants expressed frustration with marketers that over-inflate claims about their cryptographic products in an attempt to differentiate the security of their product over others. P74, a technologist at a start-up company, stated his issue with product marketing, *"If you don't write hyperbolic claims in slimy marketer-speak, you don't get in the front door even with solid solutions."*

V. Limitations

Our study has several limitations. Since there is no prior research into what is representative of the cryptographic development community, our results may not generalize to the entire population of developers and organizations who implement cryptography. One of the biggest obstacles to generalization is that the sampling frame was heavily biased towards individuals on a government-maintained cryptography mailing list who may be more likely to be aware of and interested in cryptographic standards, especially those published or endorsed by the U.S. Government. Additionally, not all of the individuals on the mailing lists met our criteria of having experience in developing products that use cryptography, as the list may include many who are cryptography researchers, individuals who represent cryptographic certification organizations, and others who have an interest in following the field, but do not develop products. This may account, in part, for the low survey response rate. The sample is also heavily biased towards U.S. organizations who are likely to have different customer sets with different requirements than non-U.S. companies. However, although not generalizable, we believe that our findings are still valuable as they begin to identify gaps and areas for future research.

In addition, individual survey questions were optional, so demographic data for certain items, like job function and organization description, are unknown for some participants. For example, the opportunity to specify a description of the participant's organization was dependent on the participant granting permission to quote responses, but only 35 out of 121 participants granted permission, with 34 providing an organizational description. Responses for open-ended questions about challenges that cryptographic products pose to organizations also had relatively low response rates (between 31 and 66 respondents), perhaps due to the effort required to answer these. Interviews may be more conducive to obtaining this type of data. To this end, we are currently conducting a follow-up interview study to delve deeper into some of the more interesting survey findings.

VI. Discussion and Future Work

Our survey was exploratory in nature, covering a wide swath of organizational practices and challenges in cryptographic development and testing. Throughout the survey, multiple participants specifically cited the difficulty, cost, and scale of cryptographic development. In this section, we discuss some of the more interesting findings that we believe may warrant additional investigation.

A. Usability

Standard libraries, free, and open source software were among the top three sources of cryptographic implementations selected by participants. Unfortunately, these types of implementations are notoriously fraught with usability issues, vulnerabilities, and inadequate constraints that may lead to developers producing insecure code [1], [8], [10]–[12], [22]. The general manager at a cryptography company supported this observation:

> *"In the market, we see a tendency to simply 'borrow' from Open Source implementations without an understanding of the task at hand. As a result, the crypto deployed is often not deployed [appropriately] which increases the risk of flawed implementations."* (P23, general manager at a cryptography company)

Another participant noted, *"Even with standardized libraries, ciphers, etc. it is very easy to implement them poorly and create side channel attacks"* (P96, architect and manager for an open-source security project). This survey supports previous work suggesting that there is a pressing need to improve the usability of cryptographic APIs and associated documentation.

Over 90% of organizations surveyed consult cryptographic standards to help them select or develop cryptographic implementations. Participants also indicated that cryptographic standards are used throughout product design, development, and testing processes. However, today's standards have no emphasis that we are aware of on the usability of programmer APIs, end-user software, or documentation. Furthermore, we have little understanding of the usability of these standards and associated documentation in and of themselves. Given that our survey demonstrates the difficulty of finding skilled professionals to do this type of work, adding usability requirements to and improving the usability of standards might result in reducing the training and domain knowledge that developers require to use cryptography. Further work needs to be done to better understand how the usability of standards can be improved to support both novices and experts.

B. Testing and Evaluating Cryptographic Implementations

Although validated algorithms are in wide use, many implementations of these algorithms are not validated. Validated or not, it is currently unclear how organizations should test their cryptographic software. Over 78% of participants said that they use test vectors, but there are many unanswered questions about these vectors. For example, what are the current shortcomings, if any, of these test vectors? How do we make these test vectors and their accompanying documentation more usable?

Less than half of respondents use more advanced testing approaches such as side-channel leakage search and formal methods. Other studies [12]–[14], [16], [17] contend that there is a lack of quality tools necessary to identify cryptographic misuse within code. However, our study, with a few exceptions, showed an overall lack of concern with respect to obtaining appropriate developer tools when participants were asked about challenges. This might indicate that developers may not understand how to properly test, evaluate, and find vulnerabilities in cryptographic code, and therefore don't realize that the tools they are currently using may be inadequate. How can the research community advance testing tools and ensure they are usable to a wide audience?

There is likewise a need to make the criteria for what constitutes rigorous testing more visible and accessible. This is evidenced by thirteen of our survey respondents who indicated that they do no formal testing, but rather perform a visual inspection of data to determine if their cryptography is operating properly—assuming, perhaps, that plaintext is easily recognizable, and that anything that is not plaintext is encrypted. An obvious problem with this approach is that there are many transformations that render plaintext incomprehensible without providing cryptographic protection, such as Base64 encoding, compression, or encrypting with a constant key.

In addition, we found that organizations often use their own cryptographic implementations, raising questions about the process by which they validate the security of these implementations as well as implications of customers using potentially non-validated, non-certified cryptographic implementations. We believe this is yet another area ripe for additional exploration.

Finally, over half of survey respondents said they use third-party testing, with most of those trying to attain some kind of formal certification to a national or international standard (e.g. FIPS or Common Criteria). However, the cost of testing products may limit participation in these programs. P103, a lead project architect at a cryptographic product development company, commented, "*Cryptographic certifications (e.g., FIPS 140 or Common Criteria) are VERY difficult, expensive, and time consuming - much more so than evaluations for other kinds of products.*" These observations point to the need for more in-depth investigation into the perceived benefit vs. cost of certification programs.

C. Evaluation Metrics

As our survey demonstrates, there are many possible ways to evaluate cryptographic implementations. But given the wide variety of choice in cryptographic implementations and the differing opinions of what is important depending on an individual's job function, without clear evaluation metrics, organizations may not choose cryptographic implementations that reflect their goals and priorities. We believe that more work needs to be done to better understand factors affecting implementation decisions and how to design tools to support these decisions.

Trust of standards is a less tangible influence not explicitly asked about in the survey, but worthy of further exploration. Several of the open-ended responses suggested potential issues with trust of standards bodies, particularly the trust of government standards by non-government organizations given concerns in recent years [23], [24]. One survey participant echoed this doubt, saying, "*lost trust in US-based standards [has] become more prominent*" (P95, software engineering manager).

D. Cryptographic Agility

Customer resistance to upgrading to new cryptographic algorithms and products emerged as a challenge to developers wanting to move towards state-of-the-art. For example, many organizations experienced problems when the Message Digest #5 (MD5) cryptographic hash algorithm was deprecated, or during the transition to Security Socket Layer (SSL) 2.0 to SSL 3.0 to TLS. Although we did not specifically ask our respondents what problems they might encounter when cryptographic algorithms

are no longer approved for use in an application, the general comments that we received on the difficulties caused by cryptographic agility and the lack of customer understanding of cryptography in general indicates that many will have problems. While the ability to use new cryptographic algorithms may be a boon to developers, algorithmic deprecation clearly causes problems that could be measured quantitatively.

E. Organizational Focus

We believe there is a gap in the literature in exploring organizational, rather than individual, practices in the development of products that use cryptography. Past studies have been useful in highlighting some of the pitfalls of including cryptography within products, but it is unclear how these apply to organizational development and testing. Are organization developer populations more likely to be skilled and use more formal methods for development since the reputation and profit of their company depends on a robust, error-free implementation? Given that 45 respondents in our survey indicated that they face challenges explaining their products to customers, it would also be interesting to explore organizational approaches to communicating the value of cryptography to non-technical audiences.

Our study serves as a first step in understanding these organizational practices and associated challenges. We are currently conducting a follow-up interview study to delve deeper into some of our survey findings, and see the potential for much future research in this area.

References

[1] M. Green and M. Smith, "Developers are not the enemy! The need for usable security APIs," *IEEE Security and Privacy*, vol. 14, no. 5, pp. 40–46, Sept 2016.

[2] P. Nguyen, "Can we trust cryptographic software? Cryptographic flaws in GNU privacy guard v1. 2.3," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Heidelberg: Springer, 2004, pp. 555–570.

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

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

[5] "National Information Assurance Partnership: Approved protection profiles," 2017. [Online]. Available: https://www.niap-ccevs.org/Profile/PP.cfm

[6] L. Church, M. Kreeger, and M. Streets, "Introducing usability to the Common Criteria," in *Proceedings of the 9th International Common Criteria Conference (ICCC)*, Jeju, Korea, 2008.

[7] 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 *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 289–305.

[8] P. Gutmann, "Lessons learned in implementing and deploying crypto software." in *Usenix Security Symposium*, 2002, pp. 315–325.

[9] S. Fahl, M. Harbach, T. Muders, L. Baumgärtner, 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*, ser. CCS '12. New York, NY, USA: ACM, 2012, pp. 50–61. [Online]. Available: http://doi.acm.org/10.1145/2382196.2382205

[10] 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)*. New York, NY, USA: ACM, 2013, pp. 73–84. [Online]. Available: http://doi.acm.org/10.1145/2508859.2516693

[11] 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)*. New York, NY, USA: ACM, 2016, pp. 935–946. [Online]. Available: http://doi.acm.org/10.1145/2884781.2884790

[12] S. Arzt, S. Nadi, K. Ali, E. Bodden, S. Erdweg, and M. Mezini, "Towards secure integration of cryptographic software," in *2015 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward! '15)*. New York, NY, USA: ACM, 2015, pp. 1–13. [Online]. Available: http://doi.acm.org/10.1145/2814228.2814229

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

[14] S. Shuai, D. Guowei, G. Tao, Y. Tianchang, and S. Chenjie, "Modelling analysis and auto-detection of cryptographic misuse in Android applications," in *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*, Aug 2014, pp. 75–80.

[15] 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.

[16] 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)*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 75–88. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30598-6_6

[17] D. J. Bernstein, T. Lange, and P. Schwabe, "The security impact of a new cryptographic library," in *Proceedings of LatinCrypt 2012*, 2012, pp. 159–176.

[18] B. Schneier, K. Seidel, and S. Vijayakumar, "A worldwide survey of encryption products," Berkman Center, Tech. Rep. Research Publication 2016-2, 2016. [Online]. Available: https://www.schneier.com/academic/paperfiles/worldwide-survey-of-encryption-products.pdf

[19] B. G. Glaser and A. L. Strauss, *The discovery of grounded theory: Strategies for qualitative research*. Transaction Publishers, 2009.

[20] D. R. Thomas, "A general inductive approach for analyzing qualitative evaluation data," *American Journal of Evaluation*, vol. 27, pp. 237–246, Jun. 2006.

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

[22] D. Lazar, H. Chen, X. Wang, and N. Zeldovich, "Why does cryptographic software fail?: A case study and open problems," in *Proceedings of 5th Asia-Pacific Workshop on Systems (APSys '14)*. New York, NY, USA: ACM, 2014, pp. 7:1–7:7. [Online]. Available: http://doi.acm.org/10.1145/2637166.2637237

[23] L. Greenemeier, "NSA efforts to evade encryption technology damaged U.S. cryptography standard," Sep. 2013. [Online]. Available: https://www.scientificamerican.com/article/nsa-nist-encryption-scandal/

[24] National Institute of Standards and Technology, "NIST removes cryptography algorithm from random number generator recommendations," Apr. 2014. [Online]. Available: https://www.nist.gov/news-events/news/2014/04/nist-removes-cryptography-algorithm-random-number-generator-recommendations