Article Title Page

User-Visible Cryptography in Email and Web Scenarios

Author Details *(please list these in the order they should appear in the published article)*

Author 1 Name:	Phillip J. Brooke
Department:	School of Computing
University/Institution:	Teesside University
Town/City:	Middlesbrough
State (US only):	
Country:	United Kingdom

Author 2 Name:	Richard F. Paige
Department:	Department of Computer Science
University/Institution:	University of York
Town/City:	York
State (US only):	
Country:	United Kingdom

Corresponding author: Phillip J. Brooke
Corresponding Author's Email: pjb@scm.tees.ac.uk

□ *Please check this box if you do not wish your email address to be published*

Acknowledgments (if applicable): N/A

Biographical Details (if applicable):

[Author 1 bio]

Phil Brooke completed his D.Phil. in Computer Science at the University of York in 1999. Subsequently, he worked as a software engineer in the security sector for two years and then five years as a senior lecturer at the University of Plymouth. He is currently a Reader in Computer Science at Teesside University's School of Computing. His main research interests involve security and formal methods.

[Author 2 bio]

Richard F. Paige is Professor of Enterprise Systems at the University of York. He leads research on model-driven engineering, agile processes and formal methods. He is on the editorial board of Empirical Software Engineering, the Journal of Object Technology, and the journal Software and Systems Modelling, and is a member of the steering committees for the ICMT, ECMFA and STAF series of conferences. He has published over 150 papers in international conferences and journals. He received his PhD in Computer Science from the University of Toronto in 1997.

Structured Abstract:

- **Purpose**

  To classify different types of "user-visible cryptography" and evaluate the value of user-visible cryptographic mechanisms in typical email and web scenarios for non-expert IT users.

- **Design/methodology/approach**

We review the existing literature, then identify user stories typical to our users of interest. We analyse the risks, mitigations of risks and the limits of those mitigations in the user stories.

- **Findings**

  The scenarios identified suggest that background, opportunistic encryption has value, but more explicit, user-visible cryptographic mechanisms do not provide any further mitigation. Other mechanisms beyond technological mitigations provide the required mitigation for our users.

- **Research limitations/implications**

  Further work should be carried out on the trust issues with trusted third parties, as they are intrinsic to global, automated cryptographic mechanisms. We suggest that deployed systems should rely on automation rather than explicit user involvement; further work on how best to involve users effectively remains valuable.

- **Practical implications**

  Deployed systems should rely on automation rather than explicit user dialogues. This follows from recognised aspects of user behaviour, such as ignoring dialogues and unconsciously making a holistic assessment of risk that is mostly mitigated by social factors.

- **Social implications**

  The user populations concerned rely significantly on the existing legal and social infrastructure to mitigate some risks, such as those associated with e-commerce. Guarantees from third parties and the existence of fallback procedures improve user confidence.

- **Originality/value**

  This work uses user stories as a basis for a holistic review of the issues surrounding the use of cryptography. We concentrate on a relatively large population (non-expert IT users) carrying out typical tasks (web and email).

Keywords: Security, cryptographic controls, email, web, legal aspects, regulation, risk management

Article Classification: General review

For internal production use only

Running Heads:

(title & authors removed for review)

Abstract

Purpose

To classify different types of "user-visible cryptography" and evaluate the value of user-visible cryptographic mechanisms in typical email and web scenarios for non-expert IT users.

Design/methodology/approach

We review the existing literature, then identify user stories typical to our users of interest. We analyse the risks, mitigations of risks and the limits of those mitigations in the user stories.

Findings

The scenarios identified suggest that background, opportunistic encryption has value, but more explicit, user-visible cryptographic mechanisms do not provide any further mitigation. Other mechanisms beyond technological mitigations provide the required mitigation for our users.

Research limitations/implications

Further work should be carried out on the trust issues with trusted third parties, as they are intrinsic to global, automated cryptographic mechanisms. We suggest that deployed systems should rely on automation rather than explicit user involvement; further work on how best to involve users effectively remains valuable.

Practical implications

Deployed systems should rely on automation rather than explicit user dialogues. This follows from recognised aspects of user behaviour, such as ignoring dialogues and unconsciously making a holistic assessment of risk that is mostly mitigated by social factors.

Social implications

The user populations concerned rely significantly on the existing legal and social infrastructure to mitigate some risks, such as those associated with e-commerce. Guarantees from third parties and the existence of fallback procedures improve user confidence.

Originality/value

This work uses user stories as a basis for a holistic review of the issues surrounding the use of cryptography. We concentrate on a relatively large population (non-expert IT users) carrying out typical tasks (web and email).

Keywords: Security, cryptographic controls, email, web, legal aspects, regulation, risk management

1 Introduction

Cryptographic mechanisms are embedded in a range of software systems, including widely used distributed applications that support Internet banking (*e.g.*, via web or smartphone applications) and online shopping. Such mechanisms are designed to provide end-users, designers and auditors with a significant degree of confidence that their communications (*e.g.*, between customer and bank, or between online shop and customer) can only be read by their intended recipients, that confidential information is protected (perhaps according to its value), and that in some scenarios interactions with security mechanisms can be audited, to help to ensure that the mechanisms are operating according to their specifications.

Cryptographic mechanisms appear in distributed applications in two flavours: those that are used explicitly and interactively by end-users (*e.g.*, PGP add-ons for email clients), and those that are *hidden*: used implicitly by end-users (*e.g.*, the HTTPS protocol in a web browser). We call the former *user-visible cryptographic mechanisms*. Such mechanisms are widespread; for example, they include those that require entering of passwords or passphrases for secret keys, dialogues related to resolution of problematic (*e.g.*, revoked, stale) SSL certificates on websites, *etc*. They are used in applications including email, web browsing, e-commerce and document management; specifically, such applications are widely used by non-IT experts in a particular social and legal context, such as in a private dwelling in the UK.

User-visible cryptographic mechanisms are applied to protect assets of the actors involved in an interaction. The question that this paper considers is whether the *investment in effort* required by end-users exploiting user-visible cryptographic mechanisms is worthwhile. More precisely, we ask whether the risks or threats that *non-expert IT end-users* are exposed to whilst trying to achieve their objectives are effectively addressed by user-visible cryptographic mechanisms.

As such, this paper analyses the use of user-visible cryptographic mechanisms in representative categories of applications — specifically email and web-based software systems. The analysis aims to assess what benefits are obtained from use of user-visible cryptography by *non-IT expert* users when attempting to achieve specific goals. Our analysis, which takes into account the social and legal context in which applications are used, is scenario-based and qualitative (Kazman et al., 1996), derived from the precise specification of user stories which endeavour to elicit the risks and threats that arise from trying to accomplish said goals. Our hypothesis is that, for non-IT expert users in scenarios using typical off-the-shelf applications, there is little to no value obtained from application of user-visible cryptography. The analysis aims to provide evidence to support this hypothesis.

1.1 Structure

First, we survey related work concerning general usability and security issues in Section 2. Then, we clarify the context and scope for our analysis, including the legal context (namely, an English and Welsh perspective) and the types of users; this is given in Section 3. We describe a *general Internet user* stereotype. We omit an analysis of the underlying technology; such a discussion can be found in Brooke and Paige (2013, section 4), which includes further discussion on legal aspects and usability.

Section 4.1 presents the analysis, starting with an initial classification of types of applications, which then drills down into two specific types of applications: sections 4.2 and 4.3 deal with risks and the use of cryptography in web and email systems respectively. Additional scenarios (omitted due to lack of space) are elaborated in Brooke and Paige (2013). We synthesise the results and lessons learned from the analysis in Section 5 before concluding in Section 6.

2 Related work

The scenario-based analysis we present is focused on software applications where cryptographic mechanisms are visible to users in different ways. Overall, the domain of our analysis is at the intersection of usability and cryptography. There has been previous research at this intersection in both general terms and in terms of specific mechanisms (such as public-key infrastructure). In this section we review the general literature on usability and security; when we discuss specific mechanisms in later sections, we include further directly relevant literature.

A classic paper at the intersection of cryptography and usability is Whitten and Tygar (2005), which concerns the ability of users to use PGP 5.0: "Our 12 test participants were generally educated and experienced at using email, yet only one-third of them were able to use PGP 5.0 to correctly sign and encrypt an email message when given 90 minutes in which to do so". More generally, Furnell and others have investigated the usability of end-user software and found continuing problems with interfaces (Furnell et al., 2006; Furnell, 2007; Ibrahim et al., 2010; Sweikata et al., 2009; Cranor and Garfinkel, 2005; Gutmann and Grigg, 2005). Some attention has also been paid to the education of users in the use of security-related software (Reid et al., 2005).

Others comment on the usability of software with cryptographic features. Kapadia (2007) remarks "I found that [OpenPGP applications] were unusable with nontechnical correspondents because it required them to install additional software", which motivates our examination of systems such as IronPort and Hushmail in section 4.3.2. We used a similar approach of server-side cryptography in support of document security (Brooke et al., 2010).

Previous work has also examined PKIs and questioned their effectiveness and usability (Gutmann, 2003; Straub and Baier, 2004). Moreover the *need* for PKIs, electronic signatures, *etc.* is not clear in practice (Mason *et al.*, 2011). Alternatives involve opportunistic encryption (Garfinkel, 2003b), key continuity management (Gutmann, 2004; Garfinkel and Miller, 2005), identity-based encryption (Shamir, 1985; Martin, 2006) and email-based identification and authentication (EBIA) (Garfinkel, 2003a).

More recently, Herley (2009) argues that users' rejection of much conventional security advice (for example, ignoring SSL certificate warnings) is rational. This is on the basis of out-of-date advice and false positive warnings against the cost (to the end-user) of acting on this information. Herley examines password rules, phishing site identification and SSL certificate warnings and comments "the burden [to the end-user] ends up being larger than that caused by the ill it addresses". Similarly, Böhme and Grossklags (2011) argue that human attention is a scare resource. They too make the point that user inattention can be rational.

3 Context and scope

As mentioned in Section 1, we will carry out a scenario-based analysis of non-expert IT users engaging with cryptographic mechanisms. Any scenario-based analysis (such as use case modelling) is carried out within a context and scope, and for a particular stereotype of user. For assessing the efficacy and value of cryptographic mechanisms (or security mechanisms more generally), the *type* of embedding application (i.e., the application in which the mechanism is included) as well as the *legal context* in which the applications are used, are relevant.

Legal context must take into account the legal jurisdiction in which the analysis takes place. Consider a jurisdiction in which there are non-existent protections for e-commerce credit card fraud; the value of cryptographic mechanisms is therefore different than a jurisdiction in which there are strong protections (*e.g.*, a limit for which the consumer is liable). Legal context thus includes the specific legal system in which the analysis takes place; this paper focuses on English and Welsh law. However, across legal systems there are a number of generic concerns that are relevant in scenario-based analysis such as this. These include:

- *Integrity concerns,* related to the rules defining the acceptability of well-formed information, such as a contract. For example, a simple email indicating agreement, or completing an online form by clicking "accept" may be sufficient to form a contract in some jurisdictions. More elaborate legal systems may require the use of "signatures", where examples include hand-written signatures, email signatures that are automatically appended, *etc.*, all the way to cryptographic digital signatures. Some legislation explicitly addresses the recognition of electronic signatures, such as the Electronic Communications Act 2000 (HMSO, 2000). A result of this is that a wide range of statements can be legally considered an "electronic signature". A thorough coverage of the legal issues surrounding electronic signatures is in Mason's book (Mason, 2012).

- *Data protection,* where legislation can impose significant requirements on how data is stored, managed, audited and accessed. For example, "appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data" (HMSO, 1998, Principle 7). The definition of "appropriate" is, of course, subject to each individual scenario. Substantial guidance exists, along with a range of standards such as the ISO 27000 series. Besides regulatory requirements, information usually has value to both individuals and businesses, regardless of the presence or absence of personal data. All this needs protecting in the traditional senses of confidentiality, integrity and availability.

We must also consider stereotypical users of systems with cryptographic mechanisms. Expert users are out-of-scope in this paper; these are users who are familiar with how to use cryptographic mechanisms, and likely how they work, and have thorough understanding of the threats and risks that the mechanisms mitigate. That is, we exclude relatively small, specialised user groups with very significant security demands. Instead, we focus on users who have a limited or non-existent threat model for internet-capable applications, specifically: domestic users with tasks such as social email, online shopping and e-banking; and office workers, using software such as office productivity applications, undertaking sensitive discussions by email, or working with sensitive data such as personal data. Thus we are concerned with a "general Internet" stereotypical user without specialist skills or needs. A common assumption to all these users is that they have basic computer skills, *e.g.*, word processing and email, but they are not IT specialists and have no need (nor interest, often) to be IT specialists.

We have highlighted the legal context and user context in which we will carry out our analysis. We must next consider the types of application that are in and out of scope. We discuss this next, and then consider two specific, widely used classes of application in much more detail.

4 Analysis

Having set the legal/user context for our scenario-based analysis, we next categorise internet-based applications in terms of their mode of exploitation of cryptographic mechanisms. We do this very broadly in Section 4.1, and then analyse two subclasses of applications in detail in sections 4.2 and 4.3.

4.1 Categories of user-visible cryptography

As discussed earlier, there are many different types of user-visible cryptography. We categorise it into three broad groups: *direct*, *indirect* and *background*.

1. The most obvious user-visible cryptography is the **direct**, elective invocation, by a user, of a cryptographic application directly, *e.g.*, PGP or GnuPG.

2. **Indirect** but still explicit, elective use of cryptography involves examples such as

- asking an S/MIME email client (*e.g.*, MS Outlook) to encrypt or sign an email;

- encrypting or signing a document in an office application (*e.g.*, MS Office, LibreOffice); or

- selecting encryption in a ZIP archive application (*e.g.*, 7-Zip).

Sometimes this interaction is a simple as a user ticking a box to select encryption and giving a password which is subsequently used (in some form) as a key to a symmetric algorithm. Other interactions, such as signing office documents, require at least a user certificate for an asymmetric algorithm or a full PKI. We distinguish this from direct user-visible cryptography by the amount of interaction required and the purpose of the software (*e.g.*, is the application purely cryptographic, or is cryptography a mechanism/feature of an application with other purposes).

3. Much cryptography occurs in the **background**: we do not consider this to be user-visible cryptography. Web browsers and email clients can automatically use SSL as discussed in Sections 4.2 and 4.3. This is implicit and should be unobservable by the user until there is a problem, such as an out-of-date or otherwise invalid certificate causes the client software to warn the user. When an application is behaving normally according to its specification, the user should be unaware of the presence of the cryptographic features.

Direct and indirect uses of cryptography normally have a direct impact on the user: the user may be required to interact with functionality that implements the cryptographic mechanisms, or may be required to interact with functionality in the form of a proxy.

For example, a signed document might not require any special interaction, yet the client software may report the state of the signature, possibly raising dialogues or showing warnings in the case where the signature is no longer valid (*e.g.*, if part of the document has been changed). In other situations, the recipient may be completely unaware of the signature (*e.g.*, an office document with an embedded signature, or a multipart signed email) or conversely, the document may be unreadable without using specialist software (such as ASCII-armoured signed emails).

There are many other examples. Encrypted emails and ZIP archives nearly always require a direct interaction with cryptographic mechanisms to provide a relevant key, usually in the form of a password or passphrase. For email, the relevant private key may already be accessible for automatic decryption, as in some configurations of MS Outlook. Additionally, explicit key management may be required on part of the users, *e.g.*, to establish communications with new users external to an organisation.

Direct and indirect uses of cryptography are the most important from a usability perspective, and there are many relevant examples of such applications that are used regularly by our stereotypical user base. We now analyse two types of such applications in more detail; further analysis are presented in Brooke and Paige (2013). As mentioned earlier, the analysis will be scenario-based: we present basic scenarios of use for the two selected categories, and use these to assess the effect of the cryptographic mechanisms on the users achieving their goals.

4.2 Web

We first examine three scenarios illustrating the use of web applications. There are representative of large classes of applications relevant to our stereotypical end-users.

Browsing a social website

Alice reads and sometimes posts on a social website, *e.g.*, Facebook or web forums. The overall risk here is low: antisocial behaviour and account hijacking are the main risks, but the assets concerned are limited, at least from Alice's perspective. A greater risk might be posed by Alice posting something she later regrets.

Buy goods via a website

Alice wants to buy something from an e-commerce site. She will necessarily use her credit card or a service like Paypal. Either way, at some point, she has to pay money in the expectation that the purchase is delivered as specified. The risks are high here: phishing, website spoofing and non-delivery of goods are the canonical examples, along with theft of payment and other details from the recipient site. However, not all are related to cryptographic aspects (*e.g.*, non-delivery of goods).

Online banking

Alice views account details and pays bills using her bank's online service. The risk: bank details have an obvious value to criminals.

These scenarios are exemplars of applications with background cryptography via HTTPS. In some cases, we can see examples of indirect cryptography, such as the use of OAuth for granting rights on social networks. We were not able to identify any common web-based scenarios involving our stereotypical end-users using direct cryptography.

4.2.1 Risks

The main risk associated with these scenarios is the compromise of login credentials: these credentials are useful to attackers for harassment/nuisance via social media, theft from online banking or misuse of credit card details. Compromised credentials can then be used to call into question the integrity of any transaction involving those credentials or to present the possibility of compromised credentials for "plausible deniability". Additionally, the re-use of credentials, even on ostensibly low-security websites permits further exploitation of credentials (BitDefender, 2010).

4.2.2 Mitigations

The typical mitigation applied to address these risk in these scenarios is to use HTTPS rather than HTTP. In each case, Alice will have to point her web browser to the correct URL: this URL might have been bookmarked from a previous visit, found via a search engine or typed in, perhaps from an advert in a newspaper, or from memory.

A related mitigation is the use of certificate authorities (CA) to bootstrap what is essentially a trust relationship. The CA uses an X.509 cryptographic certificate to assert that the server connected to by Alice is indeed the intended server. Therefore, Alice should be confident to provide her credentials and other data to this server.

4.2.3 Limitations to mitigations

It is important to observe that the effective use of HTTPS is dependent on the correct use of X.509 certificates via CAs. The main limitations arise from issues around CAs and the associated certificates.

Firstly, users are often given warnings related to certificates that are disregarded or confusing. Common advice given to users for e-commerce transactions typically includes "Check that the padlock sign is shown on your browser and that the URL includes https." Regardless, users still find it difficult to assess whether or not "a connection [to a web site] is secure" (Friedman et al., 2002). Complications include extended validation and more sophisticated phishing attacks (Jackson et al., 2007). Kirlappos et al. (2012) argue that trust seals are ineffective, and conclude that "automatic verification of authenticity" is required. Rapidly changing browser environments are also likely to confuse users; for example, Mozilla Firefox has changed its indication of secure connections several times (Shultze, 2012). Besides, warning dialogues are often disregarded (Likarish et al., 2008): we have effectively trained our users to ignore the warnings because they have to workaround problems.

Next, we can concern ourselves with compromised CAs and certificates. The CA root certificates bundles in web browsers are implicitly trusted, and this trust relationship has been highlighted as potentially problematic (Perlman, 1999). This was brought sharply into focus with the Comodo compromise in 2011 (InfoSecurity, 2011) and the more recent issues with DigiNotar (Corbet, 2011). A secondary issue to the Comodo and similar compromises concerns the limited use of certificate revocation lists and the Online Certificate Status Protocol by clients to revoke bad certificates.

One difficulty concerns naming schemes of networked computers. In one sense, this is an artefact of a global naming scheme (the DNS) and we see that the simple public-key infrastructure (Ellison, 2004) suggests local naming schemes in closed groups. But this poses difficulties for, say, the banking scenario, which requires a global naming scheme.

Key management remains a major problem. Gutmann (2003) reports that obtaining a key from a public CA "takes a skilled technical user between 30 minutes and 4 hours work". Little has changed since then, and in any case, these certificates are "low value". Local CAs using the SPKI model can more easily issue certificates for their own servers, and can ensure that centrally provisioned machines have the relevant root certificate installed. But external users do not benefit from this.

The issues above concerning key management and warning dialogues are subject to the client web browser and underlying operating system operating correctly. There is a large, well-documented range of malware which can compromise client systems some of which is delivered via web browsers or their associated plugins. As well as consequences not directly relevant to our argument (such as incorporating victim machines into botnets), they clearly compromise the integrity of the client. Thus credential stealing becomes easier and the PKI protections can be rendered ineffective. Mitigations to this include browser and other software updates (e.g., plugins); antivirus software; and the increasing use of sandboxed processes. The former often require user interaction, although some vendors are moving to a model of "pushing" updates to reduce the incidence of outdated software.

4.2.4 Summary

We might ask whether the end-users should be concerned with the limitations to the mitigations of the risks they may encounter. However, we assert that the average user cannot personally address the limitations other than by withdrawing from the activity, and due to the low likelihood of a compromise personally affecting them, is unlikely to care. In the event that they *are* affected, then their remedies are outside the cryptographic structure: the social and legal context gives them recourse for compromised credentials and the subsequent impact.

4.3 Email

Our next collection of scenarios is derived from use of standard email applications (*e.g.*, mail clients such as Outlook, or webmail). Some of the most typical scenarios for such applications are as follows.

Social email

Alice wants to email her relative or friend, Bob. The overall risk is low: the main asset is the email, and it is unlikely to be particularly valuable although potentially embarrassing. From Bob's perspective, someone pretending to be Alice is a very low risk.

Sensitive discussion by email

Suppose Alice and Bob work together and need to discuss a serious problem with a particular task. Email is one possible medium. The risks revolve around confidentiality.

Agree a contract by email

Alice agrees by email to undertake some work for a small business. The main risk here concerns non-repudiation by the business or *vice versa*. Thus it is not so much an issue of

> making the contract but one of evidencing that the contract has been properly made, *i.e.*, that the elements of consideration, intention, offer and acceptance are all present.

All these scenarios are exemplars of background cryptography via opportunistic encryption (discussed shortly). The latter two may use indirect or direct cryptography.

4.3.1 Risks

Common risks with these scenarios involve the traditional security properties. The emails should be

- confidential, *e.g.*, there is no compromise to any unauthorised third party in the first two scenarios; and
- meet appropriate non-repudiation and integrity requirements. The first two scenarios have a low risk of spoofing. The final scenario concerns the repudiation of a contract.

In both cases, the level of impact varies with the scenario. Further, where a user uses a web mail client, they inherit all the same risks and issues as web users (section 4.2); this is in addition to endpoint risks that we discuss below.

4.3.2 Mitigations

One typical way to mitigate confidentiality risks is through opportunistic encryption (Garfinkel, 2003b). A mail user agent or mail submission agent connecting to a server may use SSL/TLS to encrypt the conversation with the server. If the server requires user authentication for sending emails, then it partially mitigates the integrity risks.

Opportunistic encryption has some similar problems as for web applications as discussed previously: *i.e.*, how does the user know that they have connected to the correct server? When considering email, the relevant parts of the network infrastructure may be more trustworthy. For example, two major classes of mail server are those within a particular business and those for the user's ISP. In both cases, we should have a good level of confidence that the relevant part of the DNS is correct, at least from the client's perspective, and that regardless of the certificate, we have connected to the correct server.

Mobile users provide a complication. The argument above does not apply to a user temporarily visiting another organisation or using a hotspot as they cannot generally rely on the infrastructure to the same degree (for example, there is more delegation of DNS). This is no worse than connecting to a HTTPS web application as described earlier: there is the same reliance on, and risks involved with certificate authorities.

Our third mitigation uses direct cryptography: the classical cryptographic approaches such as OpenPGP and S/MIME. Capable users might choose to generate key pairs and use one of these cryptosystems to ensure confidentiality of their messages. However, these are, by observation, a tiny minority of the population as a whole. One barrier to adoption of these approaches is the need for the recipient to have a public key; this results in multiple attempts to create public keys on demand, *e.g.*, identity-based encryption. We discuss these shortly.

Moreover, within a particular organisation —with an assumption of a trustworthy infrastructure— the emails are already safe due to opportunistic encryption, other than at the endpoints. These endpoints are the sender's and receiver's computers. It is, of course, notable that users' desktop machines are a major entry point of malware, via the web or USB sticks. For example, McQueen (2010, slides 108–109) reported that 20% of users inserted a thumb drive found in a public place into their computer.

Our final mitigation for these scenarios concerns the use of gateways providing "transparent" solutions. These address the problem of recipients generating their own cryptographic keys. Exemplars are IronPort (Cisco, 2011) and Hushmail (Hushmail, 2011). Both can use a Java applet so that decryption occurs on the client machine. Additionally, both offer an option for processing messages on the server machine via a secure web session. In this latter configuration, these services are not significantly stronger than HTTPS as described above: this is recognised in such services (Hushmail, 2010, 2011; Singel, 2007). Some implementations send the email directly and only the decryption key is escrowed, which has some positive impact.

A positive side effect is that policy engines such as IronPort can be used to reduce the "fat-fingering" of emails by requiring that all out-of-organisation emails are subject to policy enforcement (*e.g.*, encryption, or simply disallowing some outbound traffic).

4.3.3 Limitations to mitigations

Our first two mitigations, opportunistic encryption and relying on a more trustworthy infrastructure, have the same issues as web applications with respect to certificate authorities and compromised certificates.

The use of classical cryptographic approaches brings usability issues as discussed in section 2. Additional problems concern the use of passwords and passphrases used for securing cryptographic keys. For example, some systems do not require a password after importing a PKCS12 file: the private key is accessible on demand. Thus someone with access to that desktop machine can read any email, even if it is encrypted to that particular key.

Even if a user perseveres and obtains a key for use with their email client, configuration and setup often remains challenging. Dialogues remain unintuitive for the most part. In the course of other work, we counted 8–9 steps to import an S/MIME certificate from a PKCS12 file, depending on email client.

Cryptographic interoperability of email clients is poor in contrast to general use of web browsers with HTTPS. We examined a range of email clients as listed in Table 1. ***(TABLE 1 here)***

Although S/MIME is generally well-supported natively, OpenPGP often requires plugins and these are not available for some MUAs, notably MS Outlook. This means that communities of users need to agree on the cryptosystem to be used; yet these communities are often not well-defined and have porous boundaries.

Some types of MIME message may not be displayed and verification may suffer from false negative results contributing to user confusion. They lead to the same issue that we encounter with web server certificates, where users are trained to ignore the warning messages, if they actually check the signature at all. Indeed, we speculate that it would take a user's correspondents a long time to notice if they were sent signed emails with a revoked key.

The endpoints remain a problem for mitigations based on gateways: for example, some local users of health service data receive messages via a "secure" gateway. But the data is stored locally, unencrypted. If we combine local plaintext storage with a transparent approach and implicit trust in the service provider, there seems to be little security advantage over opportunistic encryption of email or a "secure web dropbox".

4.3.4 Summary

As with web applications, we assert that stereotypical end-users face a low-risk of personal compromise. When combined with the limitations surrounding mitigations such as OpenPGP and S/MIME, they have little motivation to use them. Mitigations based on gateways offer a moderate mitigation, but can involve significant cost to the organisation and have difficulties with user acceptance.

Should the user suffer some form of compromise, they again have access to remedies through the social and legal context. For example, the third scenario involved a contract: a court could be asked to adjudicate the matter.

5 Discussion

So far, we have described a number of routine scenarios for non-expert IT end-users, all of which exploit cryptographic mechanisms in some way. We have identified risks, mitigations of those risks and limitations to those mitigations. We see two recurring themes in the analysis of the scenarios: trust in the infrastructure including the certificate authorities, and that our stereotypical users can manage the risks by resorting to the surrounding social and legal context rather than using indirect or direct cryptography.

5.1 Trust

Our example scenarios often rely on trusted third parties (TTPs): the certificate authorities. These TTPs are used to bootstrap trust when there is no prior relationship between the first and second parties. The users trust the CA to check the identity of the service provider and correctly link (for example) an offered X.509 certificate to that identity.

However, we have seen that this trust may be misplaced: in section 4.2 we noted issues with bootstrapping trust relationships (Perlman, 1999) and highlighted the Comodo and more recent compromises (InfoSecurity, 2011). In section 4.3, we remarked that some of these issues can be mitigated by the local network infrastructure, *e.g.*, to allow opportunistic encryption, if that local infrastructure is trusted.

Other attempts to fix the existing CA environment include *pinning* which whitelists public keys that are expected by a particular browser to make it harder for untrustworthy certificate chains to go undetected. More interesting is the use of multiple notaries as illustrated in the *Perspectives* project (Wendlandt et al., 2008) and the subsequent Convergence add-on/daemon (Convergence, 2011): both have users selecting notaries that they trust rather than relying on the default root CAs provided in (say) a web browser. However our earlier arguments suggest that casual users will not be willing to engage in any additional work to choose their trust relationships as there is no real improvement in their situation given the user effort required.

Despite the issues with CAs, we may ask why companies such as Verisign and Entrust amongst others can run a business selling SSL certificates. We suggest that there are two major factors:

- regulatory compliance, such as the PCI SSC Data Security Standard (PCI Security Standards Council, 2010); and
- the inclusion of their root certificates in major web browser installation packages.

For the relatively low cost per unit, an individual business will not need to consider the purchase for long; yet the vendors have a wide range of potential buyers and this is a business that scales well.

5.2 Risk management

The *risks* identified in these user stories are generally low as the impact of a breach is low, for example in social websites and social email. The lower this risk, the less justification there is for the costs —time, effort and money— for user-visible cryptography as distinct from technologies such as opportunistic encryption which operate in the background. Users perceiving a low impact, whether consciously or unconsciously, are unlikely to attempt to mitigate that risk. In Brooke and Paige (2013), we argue that there are some examples where user-visible cryptography is justified, *e.g.*, to convert accidental and inevitable loss of readable data on portable media into the less-problematic loss of encrypted data.

Other risks can be mitigated by other means. Notably, cryptographic signatures typically have no added legal value over other types of signatures (as described in section 3). This applies particularly to scenarios involving e-commerce or forming contracts. The mitigation in all these cases is that the parties have recourse to the legal systems, where courts would be asked to decide if a contract existed. A simple email without a cryptographic signature may be sufficient for a court. Chapter 8 of Mason (2012) discusses issues of liability further.

Two of the scenarios involved financial transactions (e-commerce and online banking). These have the highest risks in our examples. However, reactive monitoring systems, as exemplified by credit card companies, identify anomalous patterns of use which triggers out-of-band authorisation to the credit card holder. This monitoring, along with legal guarantees limiting the risk to the card holder, can substantially reduce the risk at least to the card holder; the merchant may take on greater risk, along with the issuing bank. The advertised guarantees to account holders and the relatively low likelihood of any particular individual becoming a victim versus the obvious convenience of online banking can reasonably account for the popularity of online banking. Lacohee et al. discuss similar points in relation to e-commerce transactions: confidence in third-party restitution, assurances in relation to guarantees, and fallback procedures should things go wrong (Lacohee et al., 2006). The issues with CAs and SSL simply do not pose a sufficient problem for these users to decline to use online banking and similar services.

We might reasonably consider that user-visible applications of cryptography have an inherent requirement for user effort, and that deployment involves consideration of training requirements. However, our end-users are trying to achieve relatively simple tasks; the computers are a means to an end and our user might view the computer as nothing more than a tool like a washing machine. Thus our argument is that in the scenarios we consider, asking for any significant user effort to understand and correctly use these cryptographic features is unreasonable and likely to result in non-conformance and inadvertent misuse.

6 Conclusions

We have examined two major groups of user-visible applications of cryptography. None of these are what would be classically considered “critical systems”. Instead, they are routine, day-to-day scenarios that illustrate issues of examined the balance of risk, value and trust.

Returning to our hypothesis, that “for non-IT expert users in typical scenarios using typical off-the-shelf applications, there is little to no value obtained from application of user-visible cryptography”, we see that despite the apparent problems, particularly those associated with endpoint security and trust (especially of CAs), the deployed systems work relatively well even though they do not typically involve direct user-visible cryptography. Our survey suggests that this is due to the presence of other mitigating factors, such as guarantees to bank account holders and recourse to the legal system, rather than any user-visible cryptography.

Further evidence in relation to the usage of direct user-visible cryptography by users considered in this work could be gathered by software installation metrics; for example, how many of the users of Thunderbird install the Enigmail plugin? These would not be perfect metrics: for example, users may install software then never subsequently make use of it. But if none of the software of a particular class is installed, the users cannot use that capability. Background (in our terminology) cryptographic capabilities can be measured more easily for some scenarios, for example, by scanning the well-known web and email ports of Internet hosts.

We conclude that any application of user-visible cryptography must make sense in that particular context. Our analysis illustrates the potential security issues are essentially peripheral to the user’s concerns as well as being perceived as low risk. As a result, we suggest that only background, opportunistic cryptography has any application in these scenarios. Even then, this is usually for confidentiality purposes, as integrity issues are not typically mitigated by cryptography in these scenarios.

References

BitDefender (2010). Bitdefender finds exposed social media credentials often provide access to email accounts. http://www.bitdefender.co.uk/NW1684-uk--BitDefender-Finds-Exposed-Social-Media-Credentials-Often-Provide-Access-to-Email-Accounts.html, last checked 29 July 2011.

Böhme, R. and Grossklags, J. (2011). The security cost of cheap user interaction. In *Proc. New Security Paradigms Workshop*.

Brooke, P. J. and Paige, R. F. (2013). The value of user-visible internet cryptography. arXiv:1303.1948.

Brooke, P. J., Paige, R. F., and Power, C. (2010). Document-centric XML workflows with fragment digital signatures. *Software Practice & Experience*, 40(8):655–672.

Cisco (2011). Cisco IronPort Email Security Appliances. http://www.cisco.com/en/US/products/ps10154/index.html, last checked 29th July 2011.

Convergence (2011). Convergence. http://convergence.io/, last checked 24th July 2012.

Corbet, J. (2011). Fraudulent ∗.google.com certificate issued. http://lwn.net/Articles/456798/, last checked 8th September 2011.

Cranor, L. F. and Garfinkel, S., editors (2005). *Security and Usability: Designing Secure Systems that People Can Use*. O'Reilly.

Ellison, C. M. (2004). SPKI/SDSI certificates. http://world.std.com/~cme/html/spki.html, last checked 29th July 2011.

Friedman, B., Hurley, D., Howe, D. C., Felten, E., and Nissenbaum, H. (2002). Users' conceptions of web security: A comparative study. In *Proc. CHI 2002*.

Furnell, S. (2007). Making security usable: Are things improving? *Computers & Security*, (26):434–443.

Furnell, S., Jusoh, A., and Katsabas, D. (2006). The challenges of understanding and using security: A survey of end-users. *Computers & Security*, (25):27–35.

Garfinkel, S. L. (2003a). Email-based identification and authentication: An alternative to PKI? *IEEE Security & Privacy*, 1(6):20–26.

Garfinkel, S. L. (2003b). Enabling email confidentiality through the use of opportunistic encryption. In *Proc. Digital Government Research*.

Garfinkel, S. L. and Miller, R. C. (2005). Johnny 2: A user test of key continuity management with S/MIME and Outlook Express. In *Proc. Symposium On Usable Privacy and Security*.

Gutmann, P. (2003). Plug-and-play PKI: A PKI your mother can use. In *Proc. 12th USENIX Security Symposium*, pages 45–58.

Gutmann, P. (2004). Why isn't the Internet secure yet, dammit? http://www.cs.auckland.ac.nz/~pgut001/pubs/dammit.pdf. AusCERT conference slides.

Gutmann, P. and Grigg, I. (2005). Security usability. *IEEE Security & Privacy*, pages 56–58.

Herley, C. (2009). So long, and no thanks for the externalities: The rational rejection of security advice by users. In *Proc. New Security Paradigms Workshop*.

HMSO (1998). Data Protection Act. (c.29).

HMSO (2000). Electronic Communications Act. (c.7).

Hushmail (2010). Using Java with Hushmail. https://help.hushmail.com/entries/245155-using-java-with-hushmail, last checked 19th July 2011.

Hushmail (2011). How Hushmail can protect you. http://www.hushmail.com/about/technology/security/, last checked 19th July 2011.

Ibrahim, T., Furnell, S. M., Papadaki, M., and Clarke, N. L. (2010). Assessing the usability of end-user security software. In Katsikas, S., Lopez, J., and Soriano, M., editors, *Proc.*

TrustBus, volume 6264 of *LNCS*, pages 177–189.

InfoSecurity (2011). Comodo certificate compromise has iranian fingerprints. http://www.infosecurity-magazine.com/view/16874/comodo-certificate-compromise-has-iranian-fingerprints/, last checked 29th July 2011.

Jackson, C., Simon, D. R., Tan, D. S., and Barth, A. (2007). An evaluation of extended validation and picture-in-picture phishing attacks. In *Proc. Usable Security*.

Kapadia, A. (2007). A case (study) for usability in secure email communication. *IEEE Security & Privacy*, 5(2):80–84.

Kazman, R., Abowd, G. D., Bass, L. J., and Clements, P. C. (1996). Scenario-based analysis of software architecture. *IEEE Software*, 13(6):47–55.

Kirlappos, I., Sasse, M. A., and Harvey, N. (2012). Why trust seals don't work: A study of user perceptions and behaviour. In *Proc. TRUST 2012*, volume 7344 of *LNCS*, pages 308–324.

Lacohee, H., Phippen, A.D., and Furnell, S.M. (2006). Risk and restitution: Assessing how users establish online trust. *Computers and Security,* 25:486-493.

Likarish, P., Jung, E., Dunbar, D., Hansen, T. E., and Hourcade, J. P. (2008). B-APT: Bayesian anti-phishing toolbar. In *Proc. IEEE International Conference on Communications*.

Martin, L. (2006). Fitting square pegs into round holes. *IEEE Security & Privacy*, 4(5):64–66.

Mason, S. (2012). *Electronic Signatures in Law*. Cambridge University Press, 3rd edition.

Mason *et al.* (2011). Response to *Digital Agenda for Europe*: Electronic identification, authentication and signatures in the European digital single market public consultation. http://www.law.ed.ac.uk/ahrc/ITTT/EU_Electronic_signature_consultation_Bileta_submission.pdf, last checked 12th March 2013.

McQueen, M. (2010). Software and human vulnerabilities (implications for protection of our critical infrastructures). In *Proc. IECON*. Tutorial slides.

PCI Security Standards Council (2010). Data security standard. https://www.pcisecuritystandards.org/security_standards/index.php, last checked 24th July 2012.

Perlman, R. (1999). An overview of PKI trust models. *IEEE Network*.

Reid, R. C., Platt, R. G., and Wei, J. (2005). A teaching module to introduce encryption for web users. In *Proc. Information Security Curriculum Development Conference*, pages 60–65.

Shamir, A. (1985). Identity-based cryptosystems and signature schemes. In Blakley, G. and Chaum, D., editors, *Advances in Cryptology*, volume 196 of *LNCS*, pages 47–53. Springer-Verlag.

Shultze, S. (2012). Firefox changes its https user interface… again. https://freedom-to-tinker.com/blog/sjs/firefox-changes-its-https-user-interface-again/, last checked 26th July 2012.

Singel, R. (2007). Encrypted e-mail company Hushmail spills to Feds. http://www.wired.com/threatlevel/2007/11/encrypted-e-mai/, last checked 29th July 2011.

Straub, T. and Baier, H. (2004). A framework for evaluating the usability and the utility of PKI-enabled applications. In *Proc. EuroPKI*.

Sweikata, M., Watson, G., and Frank, C. (2009). The usability of end user cryptographic products. In *Proc. Information Security Curriculum Development Conference*, pages 55–59.

Wendlandt, D., Andersen, D. G., and Perrig, A. (2008). Perspectives: Improving SSH-style host authentication with multi-path probing. In *Proc. USENIX Annual Technical Conference*.

Whitten, A. and Tygar, J. D. (2005). Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In Cranor, L. F. and Garfinkel, S., editors, *Security and Usability: Designing Secure Systems that People Can Use*, pages 669–692. O'Reilly.

TABLE 1**:** Email clients examined

	S/MIME	OpenPGP
MS Outlook	native	
Mozilla Thunderbird	native	Enigmail plugin
Apple Mail	native	
Alpine	native & filters	various filters