REPRINT
Ed Gerck: "Secure Email Technologies X.509 / PKI, PGP, IBE, and ZMAIL", in Chapter 12, Corporate Email Management, S. J. Krishna and Elizabeth Raju (Ed.), ICFAI University Press, 2007, pp. 171-196, ISBN 81-314-12797 (2007).

Chapter 12

SECURE EMAIL TECHNOLOGIES X.509 / PKI, PGP, IBE, AND ZMAIL

A Usability & Security Comparison

Ed Gerck, Ph.D.
ed@gerck.com

NMA, Inc.

Abstract: *Online secure delivery of documents, health records, legal signatures, credit authorizations, and goods such as music and movies are increasingly supported by regulation worldwide. However, email security has been lagging in providing suitable delivery means. This work extends a previous usability and security comparison of the secure email technologies X.509 / PKI, PGP, and IBE, to the new technology ZMAIL. The comparison is based on a list of 33 desirable secure email features and 17 shortcomings (attacks and problems), as a metric for quantitative evaluation. Usability and Security Scores are defined and take into account the use environment. For an enterprise use environment, the Usability Scores are ordered from highest to lowest as ZMAIL >> IBE >> X.509/PKI > PGP. The Security Scores are ordered as ZMAIL >> X.509/PKI > IBE >> PGP.*

Key words: secure document delivery, digital signature, online notarization, email, usability, security, public-key, X.509, PKI, PGP, IBE, ZMAIL, Outlook, Voltage, MessageGuard, Hushmail, ZSentry

INTRODUCTION

This paper extends our previous work in [Gerck05], which presented a methodology for comparing the secure email technologies X.509/PKI, PGP and IBE.

In [Gerck05], a *score card* was provided for each technology by comparing several products in the market, including Outlook® (X.509/PKI), PGP® and Hushmail™ (PGP), Voltage™ and MessageGuard™ (IBE). The results in [Gerck05] were extensively commented and peer-reviewed online (see [Gerck05]), validating the methodology.

In this paper, we expand the analysis of X.509/PKI, PGP and IBE, and include the new technology ZMAIL, represented by the product Zmail™. We also extend the methodology to quantitatively incorporate the use environment, which was qualitatively mentioned in [Gerck05].

The motivation for this work is that email is just NOT a secure method of communication.

Email, designed to be a simple text message, has a number of longstanding security shortcomings that are being increasingly exploited in mass scale. There are several reasons for this state of affairs [1].

In the 21st century, email can also no longer be that simple text message that it was originally designed to be.

Email is used between individuals and, more and more today, for business communication.

Individual use, today, includes functionality that was not foreseen when email was originally designed. For example, people would like to use email for delivering credit card authorizations, documents, health records, financial statements, legal digital signatures, and virtual goods such as music and movie files. Most of these applications require privacy and security.

Business use of email also includes additional needs, such as: mitigating risk when sharing security responsibilities between senders and recipients with different organization policies and persons involved in the communication, preserving formal work flow with structured documents, providing enforceable document release and retention policies, supporting multiple cross-organization end-points, allowing third-party verification services such as timestamps and antivirus scans, and enabling auditing.

Notwithstanding its limitations, email communication is still very attractive because of its low cost and very large user base worldwide, with more than 500 million users today, as estimated by the Radicatti Group.

Corporations, the largest market sector, have a clear need for email security. For very compelling business reasons including legal requirements, corporations need to send and receive private and secure messages between

specific endpoints [2] − and even be able to control them at the end-points. For example:

- Protecting consumer privacy is becoming a duty for organizations worldwide.
- Organizations in regulatory privacy and security compliance regimes (e.g., in the US, HIPAA and FFIEC) increasingly need to communicate with partners in a way that is not only compliant but also verifiably compliant.
- Email phishing, spam, email disclosure, email virus, and identity theft are major sources of fraud losses today.
- Organizations need to enforce their document retention policy (e.g., defining when documents expire and can no longer be used) in such a way that the recipient is not burdened by any duty or liability.

There are many possible requirements for privacy and security for email, with varying degrees of assurances, cost, and benefits. For example, in some cases regular email may be used with confidentiality disclaimers [3]. But what may work within an enterprise with well-trained personnel may not be suited for use cross-enterprise, with consumers, or with members of the public.

It is usually recognized that a secure email system should at least provide three Basic Features: message confidentiality, message integrity, and end-point authentication:

Basic Features of Secure Email:

- *message confidentiality* (only the dialogue parties are privy to the message),
- *message integrity* (the message was not tampered with), and
- *end-point authentication* (the dialogue parties have verified identities and / or credentials).

According to this classification, digitally signed email, for example, is not secure email. Even though it authenticates the sender and provides message integrity, it does not provide message confidentiality and does not authenticate the recipient. Encrypted email, when just message confidentiality is provided, is also not secure.

However, email security is not a mainstream application today − not even for corporations. What's missing is the Most Important Feature of all: Usability − including ease of use and ease of deployment.

Providing ease of use in email security has been a difficult task, still after 15 years of technology development (X.509 was released in 1988 and PGP in 1991). In a usability test done using PGP 5.0, when the test participants were given even 90 minutes in which to sign and encrypt a message, the majority of them were unable to do so successfully [WhTg].

Further, even without usability considerations, providing email security is by itself a difficult task. Email uses a store-and-forward messaging system from sender to recipient that is hard to control or even trace [4]. Secure Sockets Layer (SSL), an Internet security technology that can be applied well to secure credit card transactions, cannot secure email communications [5].

This paper compares the technologies X.509 / PKI (Public-Key Infrastructure), PGP (Pretty Good Privacy), IBE (Identity-Based Encryption), and ZMAIL (ZSentry Mail), in their application for secure email. There are several products in the market using these technologies to secure email, such as Outlook® (X.509/PKI), PGP® and Hushmail™ (PGP), Voltage™ and MessageGuard™ (IBE), and Zmail™ (ZMAIL).

The capabilities of these products in terms of usability and security depend, to a large extent, on the capability of the underlying technology. Looking to improve email security beyond current limitations, this work begins with the specifications of Usability and Security in Sections 1 (Usability) and 2 (Security).

Sections 4 and 5 present a technologically neutral metric for evaluating secure email systems that use different technologies, where desirable features are listed in Section 4 while shortcomings (problems and attacks) are listed in Section 5. For clarity, these Sections also present concise definitions for each feature and problem or attack considered, in the context of secure email.

Sections 1, 2, 3, 4, and 5 were already introduced in [Gerck05], which presented the technologically-neutral metrics and methodology for comparing the secure email technologies X.509/PKI, PGP and IBE. This work presents some new desirable features and remarks, based on previous comments and peer-review (see Blog in [Gerck05]).

Section 6 of this work uses the technologically neutral metric developed in Sections 4 and 5 to present a *score card* for each secure email technology X.509 / PKI, PGP, IBE, and ZMAIL, based on market products using the respective technologies.

In the Conclusions, the *score card* is applied to rate the usability and security offered by each secure email technology. The context of usage, which is likely to affect which technology is most suitable in each case, is also taken into account to define *Usability and Security Scores* for each technology.

The *Usability and Security Scores* for each technology can be understood as limiting factors for the evolution of products using these technologies.

1. USABILITY

The Most Important Feature of a secure email system is Usability.

In practice, users will rather use an insecure email system that is easy to use than a secure email system where even the help text seems intimidating. The secure email system has to be easy enough to use when compared with simple, familiar, regular email systems − not when compared with other secure email systems. If security is too difficult or annoying, users may give up on it altogether.

Ease of use is considered here to be a self-evident need in all email security systems. See for example, the paper by Alma Whitten and J. D. Tygar, "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0" [WhTg].

In "Why Johnny Can't Encrypt" (*op. cit.*), the authors report a number of user interface design flaws, that may contribute to security failures, and a user test demonstrating that when the test participants were given even 90 minutes in which to sign and encrypt a message, the majority of them were unable to do so successfully.

While Whitten and Tygar make some good points, this paper takes the stance that what's needed for improved usability in secure email is, first of all, *technology improvement. Technology defines an upper bound for usability.*

Usability may seem to be, inescapably, a user interface design problem (see, for example, [WhTg]). However, when the role of user interface design is seen as that of providing a language for communication between the user and the system, it becomes clear that the language cannot be more expressive than the system's technology allows it to be − even if the user has unlimited ingenuity and learning capacity. In short, if the technology does not allow it to exist, it cannot be expressed. Moreover, because communication cannot be 100% efficient, the system's technology provides an upper bound on usability − user interface design, no matter how clever and sophisticated, can at best only *reduce* usability when compared with what the technology is potentially able to offer. Additionally, improving the graphical user interface and the help dialogue in email security products seems to have reached a point of diminishing returns after almost 15 years of development.

Therefore, rather than calling for more work on yet another improved graphical user interface and more help text to guide the user through all the steps required to send and receive secure email, with expected meager if not

negative returns, what's needed is a real *Reduction* and *Simplification* of those steps at the technology level. What's needed is technology with less complexity, less steps, less need for help text, and less need for the user to learn anything.

Usability is an aggregation of properties.

Usability (including ease of use and deployment) is generally unpredictable when looking into specific, isolated features. But usability is not entirely subjective, either. In practice, usability emerges from simple, effective rules that allow complex patterns to be expressed.

In summary: Usability is technologically supported and user interface design will, at best, reduce the usability inherent in the technology. To support usability, the technology should have simple, effective rules allowing complex patterns to be expressed as desired, rather than rules that require complexity from the start.

We note that because simplicity is also a basic principle for increased security, usability and security are not in conflict with each other.

Contrary to common opinion, we find no fundamental need to balance security with usability.

Usability is, thus, viewed here not as the result of any isolated property, or as a purely subjective evaluation, but as the result of an aggregation of properties. Usability can be technologically provided by inclusion of usability features as well as by exclusion of usability problems or attacks, with *Reduction* and *Simplification* of all rules.

Sections 4 and 5 present a majority of entries that are usability related. **The most usability-relevant rows in Sections 4 and 5 are marked in bold.** Usability emerges as an aggregation of supported features and excluded problems or attacks, for each technology. The definition and evaluation of the *Usability Score* is presented in the Conclusions.

2. SECURITY

An email message needs to be protected end-to-end, so that no one can eavesdrop, tamper, fake, spoof, or even automatically scan and index information from it while it moves from sender to recipient. In addition, an email may also need to be protected at the end-point − with control features such as expiration ("self-destruct") and usage rights management.

The objective of a secure email system is, thus, more than just a secure transport, as done with a SSL secure web page. It is also about control at the end-points.

Control requires having to "do something" in order to sign or decrypt a message (i.e., operations that use private keys at each end-point).

For secure email purposes, one can have server control, client control or human control.

For example, for machine-to-machine secure communication one can have fully automatic email messages between two servers, signed and decrypted without anyone having to do anything − and this is secure because the servers *are* in control.

Could we then have fully automatic, machine-to-machine secure email for human communication? No, because this is not a secure scenario for human correspondence. The servers / clients could fake a digital signature or decrypt a message without knowledge of the author or recipient. Humans need to be in *sole control* of their private-keys for secure human correspondence, even if the computational tasks are executed (as they must be) by a machine.

Control of private keys include control of digital signature and decryption keys. The need for signing keys to be in sole control of the signer is well-known, in terms of meeting legal evidence requirements, including use and revocation of the signing key (see Note F4 at the end of Table 1). The same principle applies to decryption keys, to prevent information leakage.

Decryption control is the dual of signature control, the latter provides assurance to the recipient that the message was sent only by the specified sender while the former provides assurance to the sender that the message can be read only by the specified recipient.

A secure email system should, thus, provide human control of signing and decryption. Human control of encryption and signature verification (i.e., operations that use public keys at each end-point) is usually not relevant for security and can be provided automatically to improve Usability.

Sections 4 and 5 present the desired security features and the undesired problems or attacks, for each technology. The definition and evaluation of the *Security Score* is presented in the Conclusions.

3. AUDITING

The decision to trust a record (e.g., the source of a communication, or the name on a certificate) must be based on factors outside the assertion of trustworthiness that the record's system makes for itself [Gerck98]. Accordingly, an auditing system should be as independent as possible from what is audited. Such an auditing system should be supported by and used together with a secure email system (for a secure system without auditing is not secure). The needs of the auditing system, standard in the art, are not mentioned in this paper. For example, features F15 ("Verified Timestamp")

and F16 ("Verifiable Message Notarization") in Section 4 can be useful as auditing inputs that can be verified independently of the email system.

4. DESIRABLE FEATURES REFERENCE SHEET

The results of the study (see also [Gerck05]) highlighted a list of 33 desirable features for secure email, including authentication, encryption, key management, and message management, as shown in Table 1. Each feature is referenced as F*n (n=1, 2*…) and includes a concise definition.

Features marked with * are further explained in the Notes after the Table. Features marked in bold can considerably improve Usability by *Reduction* and *Simplification*. The context of usage is likely to affect what features are most suitable for each environment. Desirable features are considered positive points in the metric.

Desirable features that are obviously mandatory for secure email are not included, such as encoding the encrypted binary data into Base 64 text (6-bit) that is standards-compliant and suitable for email transport.

This work also does not include desirable features that do not depend on the secure email technology *per se*, such as address book management, font formatting, document styles, plug-ins, number and size of attachment files, and multiple Operating System operation.

Table 1. Desirable Features (entries in bold can considerably improve Usability)

REF.	FEATURE	DEFINITION
F1	Encryption (message confidentiality)	Scrambles data using an algorithm and a key. Should use well-known algorithms, implemented in standards-compliant fashion, and using keys with adequate length (e.g., 128-bit for symmetric encryption and 1024-bit for public-key encryption).
F2*	Decryption	Uses an algorithm and a key to de-scramble data.
F3	Message Integrity	Verifies that the message was not tampered with.
F4	Legal Digital Signature	Legal digital signature, including usual requirement for the signing key to be in sole control of signer.
F5	Key Expiration	Prevents a key from being used after its lifetime lapses.
F6	Key Revocation	Prevents a key from being used after notification.
F7	Identity Certificate	A data file that strongly (i.e., in way that cannot be forged) binds a key and an identity, usually including how and when

REF.	FEATURE	DEFINITION
		the identity was verified, the certificate lifetime, revocation information and issuer information.
F8	Private Key Not At Server	Server cannot autonomously generate or retrieve private key.
F9*	**Anti-Spoofing**	Sender and recipient can verify server directly (without machine intervention), preventing spoofing and phishing.
F10*	Sender Two-Factor Authentication (end-point authentication)	Verifies sender using two-factor authentication (what the sender received and what the sender knows).
F11*	Recipient Two-Factor Authentication (end-point authentication)	Verifies recipient using two-factor authentication (what the sender received and what the sender knows).
F12	**Message Expiration**	Prevents email from being read after its lifetime lapses.
F13	**Message Release**	Prevents email from being read before its release lapses.
F14	**Message Recall**	Prevents email to be read after recall notification to server.
F15*	**Verified Timestamp**	Date and time are defined by a third-party, verified time source.
F16	**Verifiable Message Notarization (Fingerprint)**	Third-party, verifiable notarization with a short, human-usable, message fingerprint for message authentication.
F17*	**Send Receipt**	Third-party confirmation to sender that message was sent, with timestamp, fingerprint, and envelope summary.
F18*	**Return Receipt**	Third-party confirmation to sender, with timestamp and fingerprint, regarding by whom, where, how, and when a message was decrypted.
F19	**Mobile Use Encoding**	Encodes encrypted binary data in a format suitable for cell phone (mobile) use.
F20	**Compact Encoding**	Encodes encrypted binary data in a compact format suitable for IM and archiving.
F21	**Attachment Encoding (easy decryption)**	Encodes encrypted binary data in a format allowing decryption from attachment without copy-and-paste.
F22	**HTML Encoding (very easy decryption)**	Encodes encrypted binary data in a format allowing decryption without attachment or copy-and-paste.
F23*	**Secure Web Form Processing**	Encrypts, decrypts and processes email data as a web form.
F24*	**Decryption Key Self-Revocation & Reset**	User can self-revoke decryption key and self-reset to a new key, without other human intervention, at any time.

REF.	FEATURE	DEFINITION
F25*	**Signature Key Self-Revocation & Reset**	User can self-revoke signature key and self-reset to a new key, without other human intervention, at any time.
F26*	**Decryption Key Self-Recovery**	User can self-recover decryption key, without other human intervention, at any time, based on a master key pre-defined by the user. The master key must be private, in sole control of the user and, preferably, memorable.
F27*	**Signature Key Self-Recovery**	User can self-recover signature key, without other human intervention, at any time, based on a master key pre-defined by the user. The master key must be private, in sole control of the user and, preferably, memorable.
F28*	**Send Unique**	With multiple recipients, send to each recipient uniquely.
F29*	**Protect CC**	Do not include CC'd recipients in header.
F30	**Protect Subject**	Do not include plaintext Subject in header.
F31*	**Read Once, Read Only, No Login**	Sender allows message to be read once, securely, without recipient authentication.
F32*	**Read Once, Read Only, No Registration**	Sender allows message to be read once, securely, without recipient registration.
F33*	**Read And Reply Once, No Login**	Sender allows message to be read once and replied, securely, without recipient login.

TABLE NOTES

F2: *To prevent silent surveillance and information leakage, the decryption key should be in sole control of the recipient. However, contrary to digital signatures (see F4), this is not a legal requirement today.*

F4: *Digital signatures are considered in terms of meeting legal evidence requirements, including use and revocation of the signing key. E.g., "...the usual legal definition of an electronic signature, which imposes a requirement that the signer maintain the means of signature creation under her sole control."[DA].*

F8: *Applies to both signature and decryption private keys. If the server has a user's private keys, the server could create, change, sign, or read a message without the user's cooperation, knowledge or consent. With Feature F8, the recipient is assured that the sender must have "done something" in order to sign it (signature control), while the sender is assured that the recipient actually received the message because the recipient must have "done something" in order to decrypt it (decryption control). Signing or decryption may be done by the server but if and only if the user provides the private key (has "done something"). Decryption control is the dual of signature control, the latter provides assurance to the recipient that the message was sent only by the specified sender while the former provides assurance to the sender that the message can be read only by the specified recipient. Absence of F8 does not imply key*

escrow (P1) because the private key may exist at the server and yet other parties cannot access it.

***F9**: To prevent spoofing and phishing, the user should be able to verify that the server is trusted to process the user's credentials before the server can authenticate the user (i.e., before the user completes the log in procedure). This should be provided visually and without software control in the user's computer, to prevent interference by malicious code.*

***F10 and F11:** Authentication of sender and recipient, which is a Basic Feature of secure email (see Introduction), is necessary for email to conform to end-to-end secure communication requirements. Without sender authentication, email messages can be easily spoofed. Without recipient authentication, email messages can be received by anyone.*

***F15:** The sender can easily fake the date fields in a regular email. This not only provides scam opportunities but also prevents email to be used in applications that require verified release time (email cannot be read before) or expiration time (email cannot be read after). Even though not all email requires such control, this is a valuable feature for today's email needs – such as directly supporting the sender's document release and retention policies (which may be different from the recipient's).*

***F17:** The Send Receipt provides third-party evidence that the message was actually sent, including its timestamp and authentication fingerprint (if desired by the sender), which allows later cross-verification with the Return Receipt (see F18).*

***F18:** To allay privacy concerns, the recipient should be informed beforehand that the Return Receipt will be sent back to the sender if the recipient decrypts the message. If the recipient wishes to decline to provide the receipt, the recipient should not attempt to decrypt the message. This is the same rule that postal mail follows. The Return Receipt is useful for both sender and recipient, in addition as evidence for the sender; for example, if the sender knows that the recipient read (decrypted) the email, the sender does not have to send another email or make a call.*

***F23:** Provides for secure structured messaging and processing, e.g. for document workflow automation, without using a SSL web site. For example, data can be collected securely using a web form (XML or HTML) that is sent to the recipient and flows back by secure email; the recipient is not able to edit the web form, just use it to input the requested data in the required format. The secure email client or server can include a plug-in for the XML or HTML specification (e.g., database format or style sheet); alternatively, if the XML or HTML specification is not-standardized yet, the secure email message itself can include the necessary encoding directly or by a secure link.*

***F24, F25, F26, and F27**: There should be no other human intervention necessary, besides the user herself, and no one else should be able to do it, except the user. Applies to any private key, including decryption and signature private keys.*

***F28 and F29:** To prevent disclosure of any but one message recipient for each message, providing communication confidentiality (in addition to message confidentiality).*

***F31 and F32:** If the Sender allows it, the Recipient is not required to login (or register) in order to read the message once (reply may require login). After the message is accessed once, any second attempt shows a notice that the message was already accessed once and can only*

be read again after login or registration. This method is verifiably secure. If the recipient can prove to herself that the message was not read before (by not seeing the notice), she knows that the message contents are, indeed, secret. Once the Recipient reads the message, the message is further protected so that a login is required prior to reading it again. The system should have logs that can be used to identify the IP number and other access data, to investigate and possibly report any unauthorized activity.

F33: *If the Sender allows it, the Recipient can read and reply once without login (or registration). See Note on F31 and F32. This method is verifiably secure.*

5. PROBLEMS / ATTACKS REFERENCE SHEET

The results of the study (see also [Gerck05]) highlighted a list of 17 problems and attacks for secure email, including access to keys, key management, key revocation delay, and message management, as shown in Table 2. Each shortcoming is referenced as P*n (n=1, 2*…) and includes a concise definition.

A secure email system that has problems will likely be used badly (insecurely) or not at all − which opens room for attacks.

Shortcomings marked with * are further explained in the Notes after the Table. The entries in bold correspond to Usability problems. **Absence** of these problems can considerably improve Usability, by *Reduction* and *Simplification*. The context of usage is also likely to affect what problems or attacks are most important.

Shortcomings that are obviously too severe for secure email, or do not depend on the secure email technology *per se*, are not included, such as sending the password with the message, using keys that are too short, or using weak encryption algorithms.

Table 2. Problems and Attacks (entries in bold can considerably impair Usability)

REF.	PROBLEM / ATTACK	DEFINITION
P1*	Private Key Escrowed At Server	Server can autonomously provide or use private key.
P2	Break Private Key Protection	Data protection of private key can be broken locally or at the server with much higher probability than a brute-force attack. (e.g., dictionary attack)
P3	Break Policy Protection At Server	Data protection of policy and privileges of users can be broken at the server with much higher probability than a brute-force attack. (e.g., dictionary attack)

REF.	PROBLEM / ATTACK	DEFINITION
P4	Weak Authentication Accepted	Accepts username / password authentication, which has a vulnerable password file and can be broken with much higher probability than a brute-force attack. (e.g., dictionary attack)
P5*	Server Spoofing	A server that mimics a legitimate website to lure users into disclosing confidential information.
P6*	Unverified Sender's Email Address	The "From:" header of the email (possibly other headers) are set to a reputable email address, to lure the recipient to read and act on the email. E.g., using the email address of a friend, the user's employer, a bank or a government agency.
P7	Phishing (Email Fraud)	The email appears to be from a well-known entity but is not. Simply clicking the link may subject the user to background installations of key logging software or viruses.
P8*	"Lunchtime" Attack	The attacker (who can be a secretary, technician, customer, etc.) can sneak into the manager's office for a few minutes while he or she is away for lunch, and use that person's computing resource.
P9	**Key Management**	Key issuance, key certification, key revocation and key distribution.
P10*	**Key Revocation Delay**	Revocation is not immediate.
P11	**Lack of Centralized Key Revocation**	Lack of a single location to post revocation notices, resulting in different revocation status potentially being posted for the same key, at the same time.
P12*	Open Message Headers	Email message headers containing cleartext information that cannot be protected, e.g., with a list of CC names, addresses and the Subject.
P13	**Must Pre-Enroll Recipients**	Recipient must register before the message can be sent.
P14	**Must Register To Read**	Recipient must register before the message can be securely read.
P15	**Must Register To Reply**	Recipient must register before the message can be securely replied.

REF.	PROBLEM / ATTACK	DEFINITION
P16	**Must Send Own Certificate**	Sender must make her own key certificate available to recipient before recipient can send a message.
P17*	**Requires Common Root Of Trust**	Sender and recipient must a priori trust a common root for security services.

TABLE NOTES

P1: *Applies to either signature or decryption private keys. The server has the user's private key (i.e., F8 does not apply) and can provide it to other parties. Key escrow is not inherent to X. 509 / PKI or PGP technology. Key escrow is inherent to IBE technology, with the PKG (Private Key Generator). An IBE server is able, therefore, to decrypt or sign any messages for any user of the IBE system. Requiring the user to check with a PKG before reading a message makes the use of multiple PKGs much more difficult, unless they can be convinced to work together, a hard problem for competing businesses. Constant checking with a single PKG also makes traffic analysis much easier. Even if the attacker cannot decrypt the message which was sent, if the attacker can monitor the central PKG (with a single administrative order, a rootkit or a man-in-middle attack), everyone's private keys can be obtained. [8]*

P5: *This attack can happen even with SSL (Secure Sockets Layer) using 128-bit encryption, and two-factor authentication (e.g., SecurID does not authenticate the server).*

P6: *With PGP, users can create keys and sign PGP certificates and for any email address they want, without necessarily owning or operating that email address.*

P8: *The user is not trusted to correctly perform security actions that could prevent this attack (such as engaging a screen lock, removing a hardware token, or locking the door). Additional security systems (such as a screen lock that is engaged automatically when the user leaves the room), which are not part of secure email technology per se, are not considered here.*

P10: *There may be a significant delay between the notification to revoke and the actual revocation; however, as long as the potential duration of the delay is known by the recipient, he/she can take that into account. See also [7] and [GerckPK].*

P12: *If email message headers are recorded and stored, everyone's communication patterns can be easily seen and sensitive subjects sorted out.*

P17: *Trust is understood as reliance; more precisely, trust is qualified reliance on information, based on factors independent of that information [Gerck98]. A common root of trust is required if at least one of those factors (upon which trust is based) is common to both parties (sender and recipient). A common root of trust is required for X.509 / PKI and IBE, not necessarily required in PGP, and not required with ZMAIL:*

- In X.509 / PKI, the sender and recipient must a priori trust each other's CA for issuing and revocation information of their respective subscribers, even when cross-certificates and bridge CAs are used (i.e., in addition to the requirement of a trusted path between the certificates). For example, the sender and the recipient must trust the recipient's CA NOT to have a large, unannounced delay between receiving a certificate revocation request and posting a certificate revocation notice that the sender can verify before sending a message using the certificate.

- In IBE, the sender must a priori know the system parameters of and trust the key server (the PKG, Private Key Generator) used by the recipient. If the trust is broken in any of these cases, system security breaks as well. For example, in IBE, if the PKG is a rogue key server, the private key may be provided to other parties in addition to the recipient (breaking the sender's message confidentiality), without the recipient's cooperation or knowledge, and in spite of best efforts by the recipient to safeguard the private key.

- In PGP, however, even though the sender and recipient must a priori trust each other's key signers (the web of trust), in practice, PGP users verify keys out of band (e.g., by phone call) with each other, not through the web of trust – eliminating the need for a Common Root Of Trust because each key can be self-signed.

- With ZMAIL, the sender and the recipient develop their own keys by authenticated key-agreement, eliminating the need for a Common Root Of Trust.

6. SCORE CARDS OF SECURE EMAIL TECHNOLOGIES (PRODUCTS)

As discussed in Sections 1 and 2, the capabilities of secure email products Outlook® (X.509/PKI), PGP® and Hushmail™ (PGP), Voltage™ and MessageGuard™ (IBE), and Zmail™ (ZMAIL), in terms of usability and security depend to a large extent on the capability of the underlying technology.

Rather than comparing one secure email product with another, the methodology used in this work looks into the technological limits for each technology choice, in terms of usability and security, as a *best case scenario* in terms of products today and also in terms of product development.

First, this work established a technologically-neutral metric of positive and negative performance points using Sections 4 and 5.

The features F1 (Encryption), F2 (Decryption), and F3 (Message Integrity), which are obviously mandatory for comparing secure email systems, are not included.

Next, for each main market product using the technologies X.509 / PKI, PGP, IBE, and ZMAIL, this work asks two questions:

1. what are the product's capabilities in terms of the list of desired features listed in Section 4 (excluding F1, F2, and F3), and
2. what are the product's shortcomings in terms of the list of attack and problems listed in Section 5.

Desired features as well as problems and attacks are marked respectively as "plus" (+) and "cross" (x) in Table 3, for each product using the technology. As more products are scanned, this work takes the *best possible scenario* for each technology. A plus point is awarded if the feature is present in at least one product using that technology, while a cross point is not

marked if at least one product using that technology does not have the shortcoming. Taken together, the scores can be seen as a *score card* for each technology.

Each *score card* column in Table 3 measures that technology's best scenario to support secure email, based on the entire capability of market products that use the technology. The entries in bold in Table 3 correspond to potential improvements in Usability, with inclusion for features and exclusion for problems / attacks. *Usability is an aggregation of all these properties, for each technology.*

Finally, the use environment is also taken into account. The weights, given in the second column of Table 3 and explained in the Table Notes, reflect typical operational relevance of each feature and each problem and attack in an enterprise environment. Any other use environment can be applied by changing the weights. For example, users in a consumer environment are subject to a different mix of desired features, problems and attacks when compared with users in an enterprise environment. Table 3 includes the weighted Security and Usability totals for Features and Shortcomings, in an enterprise environment.

The notes are further explained after the Table. See Sections 4 and 5 for Reference definition.

Table 3. Score Cards (entries in bold can considerably impact Usability)

REF.	WEIGHT*	DESCRIPTION	X.509 PKI	PGP	IBE	ZMAIL
		Security Features (weighted) **Usability Features (weighted)**	**38** **15**	**31** **13**	**27** **15**	**61** **38**
F4	3	Legal Digital Signature	+			+
F5	5	Key Expiration	+	+	+	+
F6	3	Key Revocation	+	+		+
F7	1	Identity Certificate	+	+	+	+
F8	5	Private Key Not At Server	+	+		+
F9	**1**	**Anti-Spoofing**	3.1	3.1	3.1	+
F10	3	Sender Two-Factor Authentication	+	+	+	+
F11	3	Recipient Two-Factor Authentication	+	+	+	+
F12	**3**	**Message Expiration**	+	+	+	+
F13	**1**	**Message Release**	3.1		+	+
F14	**1**	**Message Recall**				+

REF.	WEIGHT*	DESCRIPTION	X.509 PKI	PGP	IBE	ZMAIL
F15	**3**	**Verified Timestamp**	+		+	+
F16	**1**	**Verifiable Message Notarization**				+
F17	**3**	**Send Receipt**				+
F18	**3**	**Return Receipt**				+
F19	**1**	**Mobile Use Encoding**				+
F20	**1**	**Compact Encoding**		+	+	+
F21	**1**	**Attachment Encoding**	+	+	+	+
F22	**1**	**HTML Encoding**				+
F23	**1**	**Secure Web Form Processing**				+
F24	**1**	**Decryption Key Self-Revocation & Reset**	+	+		+
F25	**1**	**Signature Key Self-Revocation & Reset**	+	+		+
F26	**3**	**Decryption Key Self-Recovery**	+	+	+	+
F27	**3**	**Signature Key Self-Recovery**	+	+	+	+
F28	**1**	**Send Unique**				+
F29	**1**	**Protect CC**				+
F30	**1**	**Protect Subject**				+
F31	**3**	**Read Once, Read Only, No Login**				+
F32	**3**	**Read Once, Read Only, No Registration**				+
F33	**3**	**Read And Reply Once, No Login**				3.1
		Security Shortcomings (weighted) **Usability Shortcomings (weighted)**	**41** **33**	**53** **36**	**34** **11**	**0** **0**
P1	10	Private Key Escrowed At Server		3.2	x	
P2	5	Break Private Key Protection	x	x	x	
P3	5	Break Policy Protection		3.2	x	
P4	1	Weak Authentication Accepted		3.3	3.3	
P5	1	Server Spoofing	x	3.2	x	
P6	5	Unverified Sender's Email Address		x		

REF.	WEIGHT*	DESCRIPTION	X.509 PKI	PGP	IBE	ZMAIL
P7	5	Phishing (Email Fraud)		x		
P8	1	"Lunchtime" Attack	x	x	x	
P9	**10**	**Key Management**	x	x	3.4	
P10	**1**	**Key Revocation Delay**	x	x 3.6	x 3.5	
P11	**3**	**Lack of Centralized Key Revocation**		x	x	
P12	1	Open Message Headers	x	x	x	
P13	**10**	**Must Pre-Enroll Recipients**	x	x		
P14	**3**	**Must Register To Read**	x	x	x	
P15	**3**	**Must Register To Reply**	x	x	x	3.1
P16	**5**	**Must Send Own Certificate**	x	x		
P17	**1**	**Requires Common Root Of Trust**	x	3.7	x	

TABLE NOTES

**Weights: The weights given in table 3 reflect typical operational preferences in an enterprise environment where, for example, the need to pre-enroll recipients (P13) is a major usability shortcoming in sending secure email to customers and partners (who are not members of the organization) when compared with a "Lunchtime Attack" (P8) that is prevented by auditable internal policies.*

(3.1) Technically possible; not currently offered.

(3.2) Problem / attack for PGP Universal and Hushmail.

(3.3) Potential problems / attack, when a system accepts both weak and strong authentication to grant user access, the other party does not know which one was used. With X.509 / PKI, it is possible, based on factors such as certificate class, for the other party to verify the security policy that was used to grant user access.

(3.4) Potential problems / attack, as it requires issuance and distribution of expiration parameter if key expiration (F5) is added.

(3.5) Because IBE has no key revocation, a compromised private key cannot be terminated when desired. The revocation delay is equal to the time remaining for key expiration, which may be undefined. This implies a very large or even undefined delay for key revocation.

(3.6) In PGP, certificate revocation is done by the authenticators themselves in a happenstance pattern, with an unspecified delay. There is no guarantee if and when the revocation information is up-to-date. In fact, many PGP public keys are simply abandoned in public repositories after the user forgets the passphrase for the private key. See [Gerck99].

(3.7) In practice, PGP users verify keys out of band (e.g., by phone call) with each other, not through the web of trust – eliminating the need for a Common Root Of Trust; each key can be self-signed. However, if the sender and recipient use the web of trust, they must a priori trust each other's key signers – which requires a Common Root Of Trust.

CONCLUSIONS

The main motivation for this work is to rationally develop metrics for quantitatively comparing secure email products that use different security technologies, such as Outlook® (X.509/PKI), PGP® and Hushmail™ (PGP), Voltage™ and MessageGuard™ (IBE), and Zmail™ (ZMAIL).

The first result is a technologically-neutral set of 33 desirable secure email features and 17 shortcomings (problems and attacks), given in Tables 1 and 2. Because this first result is technologically-neutral, it can be applied as a neutral metric to rate different technologies.

More important than their pros and cons regarding security and privacy of email communication, usability was considered the *Most Important Feature* of a secure email system.

The second result is the set of *score cards* given in Table 3, for the email security technologies X.509 / PKI, PGP, IBE, and ZMAIL. The second metric was obtained by applying Tables 1 and 2 (the first result) to rate secure email products that use each technology. The bold scores have higher impact regarding usability.

The third result takes into account the use environment, not just the technology, by considering use-environment weights while tallying the features and shortcomings of each technology. For an enterprise environment (with weights set in Table 3, second column), it is specified by the four *weighted tallies* given in bold in Table 3 for each technology: *Security Features*, *Security Shortcomings*, *Usability Features*, and *Usability Shortcomings*.

The fourth result can now be calculated using the third result, as the *Usability and Security Scores*:

- *Usability Score:* Usability Features – Usability Shortcomings
- *Security Score:* Security Features – Security Shortcomings

Using the *weighted tallies* given in Table 3, for an enterprise use environment, the *Usability Score* and the *Security Score* are given in Table 4 below for each technology.

Table 4. Usability and Security Scores for an enterprise use environment

RESULTS	**X.509 / PKI**	**PGP**	**IBE**	**ZMAIL**
Usability Score (max. 41)	-18	-23	4	38
Security Score (max. 64)	-3	-22	-7	61

For an enterprise use environment, the usability difficulties with products using X.509/PKI and PGP are readily apparent from the Table above. The ZMAIL technology receives the highest Usability and Security

Scores. IBE has the second highest Usability Score. PGP has the least Usability and Security Scores.

Because any feature or problem / attack may receive different weights in a particular use profile, readers can pick and choose what they need for each use environment. Readers can make their own subset of the *score cards*, as well as calculate different *Usability and Security Scores*, valid for their needs, budget, usage and other factors.

For example, PGP is mostly used in a private use environment, where key revocation, the need to pre-enroll recipients, and other shortcomings would not be relevant. In that environment, PGP would receive a higher Usability Score than X.509/PKI.

One can also rate individual products, including products using different technologies, by comparing their *score cards*. For further quantitative product evaluation, the check mark can be changed to a product-specific grade.

Based on the results, the paper also finds that usability and security are not necessarily in conflict with each other. Contrary to common opinion, there is no fundamental need to balance security with usability. This was expected from Section 1, if simplicity is used as a basic design principle to achieve both Usability and Security.

ACKNOWLEDGEMENTS

The author acknowledges comments and contributions in private and public comments, also from readers of the previous work, including Guido Appenzeller, James A. Donald, Lars Eilebrecht, Dino Esposito, Ian Grigg, Philipp Gühring, Richard Guida, D. Gustafson, Simon McMahon, Vernon Neppe, Mike Norden, Andrew Patrick, Ralph Senderek, Einar Stefferud, Bill Stewart, Michael Ströder, Brad Templeton, and Lynn Wheeler.

This paper does not intend to cover all the details of the technologies reported, or all the variants thereof. No criticism is to be construed from this work, which respects all the apparently divergent efforts found today on the subjects treated. Products, individuals or organizations are cited as part of the fact-finding work needed for this paper and their citation constitutes neither a favorable nor an unfavorable recommendation or endorsement.

NOTES

[1] Email communication is much like anonymous postcards, answered by anonymous recipients. Anyone can send an email in your name or using your email address. Every email or attachment you send over a computer network is copied (and perhaps even backed up) on many different computers, without your

explicit knowledge or consent. However, email messages, open for anyone to read − and even write in them − are expected to carry secure and private messages between specific endpoints. The Internet, as an open network with open participation, has also increased the need for email security far beyond the threat model considerations envisioned when email was first specified, more than a generation ago, for a closed Internet with vetted participants (ARPANET). Today, the Internet has more than one billion users [Morgan Stanley Research, 2005]. A second billion users is expected to follow in the next ten years, bringing a dramatic change in worldwide security and usability needs.

[2] While email privacy and security encompass many possible needs, in general any email that can be read or used without authorization present a liability. For example, a corporation's reasoning behind a contract negotiation might be harmful if revealed even after the contract is signed. In practice, however, users are neither careful enough in selecting email content to prevent any harmful disclosure when they send email, nor assiduous in protecting all email messages they receive.

[3] In the event that regular email has to be used in a business or legal communication, content should be limited considering that disclosure to third-parties cannot be avoided and confidentiality disclaimers should be used to state that fact. The Missouri Bar Disciplinary Counsel, for example, requires all Missouri attorneys to notify all recipients of e-mail that:

(1) e-mail communication is not a secure method of communication;
(2) any e-mail that is sent between you and this law firm may be copied and held by various computers it passes through as it is transmitted;
(3) persons not participating in our communication may intercept our communications by improperly accessing your computer or this law firm's computers − or even some computer unconnected to either of us that the e-mail may have passed through.

[4] Email is not a point-to-point message between one client and one server (as a web browser viewing a web page at a server). Email is a store-and-forward message from one client to another client, with possibly several independent servers, routers, caches, buffers, content analyzers, human agents, traffic analyzers, monitors and storage devices in-between the two clients, all acting at different times and with possibly long delays. Store-and-forward supports availability, reliability and anonymity for email. Anonymity or pseudonymity, for example, cannot be achieved if there is a direct connection from the sender to the recipient because it can be traced. For strong anonymity or pseudonymity, email messages can use anonymizing remailers with random latency store and forward.

[5] SSL provides for encryption between two communicating points, such as a client application at a desktop PC and an application at a secure Internet server, usually authenticated at the server end only. Data can be transmitted using SSL over the Internet in point-to-point, two-way secure communication, encrypted at the sending point and decrypted at the receiving point. However, SSL, by itself, cannot protect an email message from client to client [4], and thus cannot prevent spam, spoofing, phishing and pharming either − all of which affect email security.

[7] For X.509 / PKI the delay betwen a certificate being revoked and the actual posting of the revocation can be quite small, through the use of OCSP or other real-time discovery technologies. However, the delay between the CA receiving from the subscriber a notice to revoke and the actual posting of the revocation can

be quite large and even, usually, unspecified by the CA (e.g., to reduce liability). See [GerckPK].

[8] Key escrow is a backdoor to decrypt a message without the author's or recipient's cooperation or knowledge; if signing keys are included, anyone's digital signature can be forged. Businesses, to avoid the key escrow backdoor liability and yet prevent problems if the proverbial bus hits an employee, may use message escrow (e.g., a secure email copy that can be decrypted by selected persons in administration) for sensitive communications. For policy or law enforcement purposes, communication information (such as routing, IP numbers, email addresses, file size, time, and frequency of use) can also be used.

REFERENCES

[Gerck98] Ed Gerck, "Trust Points", in Digital Certificates: Applied Internet Security, J. Feghhi, J. Feghhi and P. Williams, Addison-Wesley (1998), ISBN 0-20-130980-7. Online version at http://mcwg.org/trustdef.htm

[Gerck99] Ed Gerck, "Overview of Certification Systems: X.509, CA, PGP and SKIP", Black Hat Conference, 1999, in http://nma.com/papers/certover.pdf.

[GerckPK] Ed Gerck, "Certificate Revocation Revisited, Internet X.509 Public Key Infrastructure". IETF PKIX Working Group, draft-gerck-pkix-revocation-00.txt, in http://www.faqs.org/ftp/pub/internet-drafts/draft-gerck-pkix-revocation-00.txt

[Gerck02] Ed Gerck, "Trust as Qualified Reliance on Information, Part I", The COOK Report on Internet, Volume X, No. 10, January 2002, ISSN 1071 - 6327, in http://nma.com/papers/it-trust-part1.pdf.

[Gerck05] Ed Gerck, “Comparison Of Secure Email Technologies X.509 / PKI, PGP, and IBE”, published online in http://email-security.net/papers/pki-pgp-ibe.htm, with public discussions in the Blog http://email-security.blogspot.com/.

[RA] Ross Anderson, University of Cambridge Computer Laboratory, UK.

[WhTg] Alma Whitten and J. D. Tygar, "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0", in http://www.gaudior.net/alma/johnny.pdf

BIOGRAPHY

Ed Gerck received his doctorate in physics (Dr.rer.nat.) from the Ludwig-Maximilians-Universitaet and the Max-Planck-Institut fuer Quantenoptik in Munich, Germany, 1983, with maximum thesis grade ("sehr gut"). He also has titles of Electronic Engineer (1977) and Master of Science (1978) from the Instituto Tecnologico de Aeronautica (ITA/CTA), Brazil.

Ed Gerck work in information security and election integrity received worldwide press coverage by The New York Times, Le Monde, O Globo, Forbes, CBS, CNN, Business Week, Wired and USA Today.

Publications and areas of interest include information security *(information theory, secure email, trust as qualified information in human-, machine-, and heterogeneous-based communication processes, online voting,*

usability, convenience in Internet security applications), programming *(secure programming, Java, javascript, PHP, x86/87 assembler)*, physics *(quantum physics, high-power lasers, laser applications)*, and mathematics *(cryptography, Clifford algebra, topology, numerical methods)*. More information at gerck.com .