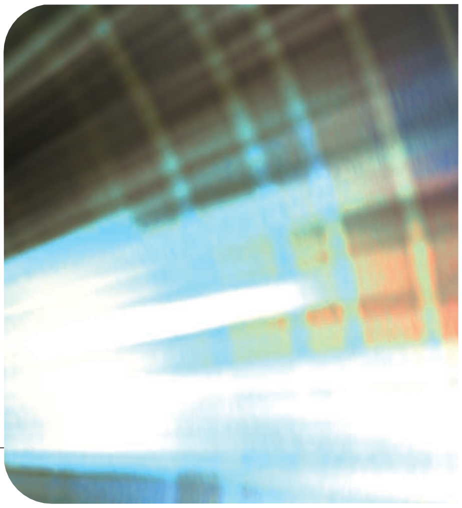
Email-Based Identification and Authentication: An Alternative to PKI?

Email-based identification and authentication is an emerging alternative to public-key infrastructure. It overcomes many problems inherent with traditional authentication techniques, such as social security numbers, and provides functional security when used within a limited context.



Simson L. Garfinkel
Massachusetts Institute of Technology

For more than a decade, businesses, governments, universities, and other organizations have developed and deployed identification–authentication systems based on public-key infrastructure (PKI). But despite this strong institutional support, an alternative system for identification and authentication organically evolved, improved, and spread during recent years. This identification–authentication regime is not based on public-key cryptography, but instead on the ability to receive email sent to a particular address.

In this article, I argue that despite some security shortcomings, email-based identification and authentication (EBIA) is a reasonable approach for many current commercial and government applications. EBIA provides a better match to the usability, privacy, autonomy, resiliency, and real-world business requirements than PKI technology. Today, even sensitive applications that let us enter into binding business agreements worth thousands of dollars (for example, on eBay) and electronically transfer money between bank accounts (for example, with PayPal), use EBIA. Here, I analyze its advantages and weaknesses, discuss best practices for its continued use, and show how EBIA might evolve into a system with stronger security properties. The "Related work" textbox on page 24 describes other PKI alternatives in progress.

Identifiers and identity theft

Personal identifiers typically are names, symbols, or codes that represent a human being. Identifiers can be contextually or globally unique: There is only one George Bush who lives at 1600 Pennsylvania Avenue in Washington, D.C., but there are two people named George Bush in the New York City telephone directory and another 15 elsewhere in New York state.

Sometimes different people can use the same identifier—a family can share a telephone number, for example. Other applications require singularly unique identifiers. In 1936, the Social Security Board adopted the nine-digit social security number (SSN) system to track the earnings of different Americans with the same names. The 1935 Social Security Act required tracking each American's earnings through his or her employment lifetime because it based, in part, retirement benefits on lifetime earnings (see "The History of Social Security;" www.ssa.gov/history/). Thus, while two people living today in New York City have the George Bush name, each of them should have a unique SSN. Moreover, those numbers should be different from that of the George Bush living on Pennsylvania Avenue in Washington, D.C., and every other person cataloged in the social security system.

Universal identifiers, which the SSN has become, are identifiers used simultaneously by different organizations. But not all universal identifiers were designed with this purpose in mind. The SSN evolved into a universal identifier as various government agencies began to use it in preference to numbers that they could issue. It was cheaper for the federal government to use pre-existing SSNs as military serial numbers, then as federal employee numbers, and, finally, as taxpayer identification numbers, than it was for all other bureaucracies to develop and maintain their own identification regimes.

But the SSN is a poor universal identifier. It lacks security features such as a check digit (to detect typographical errors) and a large space of unused codes (to decrease the likelihood that a randomly-chosen number matches a real

SSN). Nevertheless, public and private sector organizations found it easier to use SSNs than to develop a new broadly recognized system. This continued use of SSNs is a growing problem for American consumers and businesses, as it dramatically increases the risks of identity theft.[1]

Identity theft is endemic in the United States today—as many as 27.3 million US adults over the past five years have experienced some form of it[2]—because many organizations treat the SSN as both an *identifier* and an *authenticator.* That is, organizations use the SSN to identify a person, and they use knowledge of a person's SSN as a kind of proof of identity. Put colloquially, many financial organizations seem to believe that if I know your SSN, then I must be you.[3]

There are, of course, many ways for an identity thief to learn their victims' SSNs: intercepting paper mail, accessing employment records, ordering a credit report, or querying an online database. Although credit-reporting agencies require consumers to provide their SSNs when they obtain their own credit reports, "look-up" services can determine a person's SSN by knowing only a name and address or date of birth.

Many Web sites, email services, and other online systems rely on traditional user names and passwords for identification. A *user name* identifies an individual in the context of a server; users can prove their identities to servers by providing *passwords*, which only the user and service share. But user names and passwords create problems similar to SSNs: Users must disclose their passwords to prove their identities.

This process can be a significant problem for people who use the same user name and password at many different Web sites. In these cases, a password compromised at one location can have many reverberations. Spoof Web sites and email messages also can trick users into revealing their passwords.

PKI goals and pitfalls

Starting in the mid 1980s and continuing through the 1990s, PKI advocates sought to replace traditional identifiers—such as names and SSNs—with a new kind of identification system based on the ability to perform a cryptographic operation with a specified private key. Because only a single, specific individual can access that key—the theory goes—the ability to perform the cryptographic operation proves an individual's identity.

For example, if Alice wishes to use PKI to prove her identity to a remote Web site, she must engage it in a two-way protocol. Schematically, the Web site sends Alice a randomly generated number. Alice signs this number using her private key and then sends the signed number back to the Web site.

Of course, the remote Web site probably doesn't know that Alice's key actually belongs to Alice and not to somebody else. So, Alice sends with the signed number a copy of a digital certificate that contains her name, her public key, and, possibly, some other identifying information. This certificate is itself signed by some certificate-granting authority (CA) that Alice and the Web site have agreed to trust. This entire process is fully automated by the client-side digital certificate facilities present in the Secure Sockets Layer protocol (SSL) or Transport Layer Security (TLS) protocol.[4]

Client-side SSL certificates have been available commercially in the United States since VeriSign started selling them in 1996.[5] Early PKI proponents hoped that the US government might mandate PKI's use for every US citizen—or at least for every citizen seeking to do e-business with the government. Because business models of companies such as VeriSign involved selling certificates to end users for a US dollar or more, for investors, the prospect of a massive contract to supply the government seemed like the proverbial pot of gold at the end of the PKI rainbow.

But PKI wasn't just a get-rich-quick scheme. PKI was a theoretically sound way to prove identity via the Internet. PKI's big advantage over user names and passwords is that it lets individuals identify themselves in a way that does not itself compromise their actual identities. For example, the Massachusetts Institute of Technology (MIT) has a campus-wide PKI system based on the MIT Certificate. MIT's internal CA issues these certificates to students, faculty, and staff. Using an MIT Certificate, students can view their grade reports, register for classes, and even access MIT-licensed digital library resources from off campus—even from other countries. But the student's password is never sent to a Web site to verify the student's identity. Instead, students prove their identities using either the Microsoft Internet Explorer or Netscape Navigator Web browsers, a private–public key pair generated on their computers, and digital certificates that bind the students' status and identities to the corresponding public keys. This allows relatively unsecure servers to provide service to those off campus: even if the server is compromised, the student's digital identity can never be hijacked.

Another advantage of PKI is that a smart card or similar device can store the user's certificate and corresponding private key. In practice, few people use this added security option because smart cards and readers are not widely deployed. Instead, most clients store their private keys on their hard drives, sometimes with encryption—which necessitates entering a pass phrase to access the private key—but frequently without it. PKI proponents also claim that their systems provide *nonrepudiation*: that is, a digital signature made with a private key should have the legal standing of an ink-written signature on a legal document. PKI advocates have successfully passed legislation, both in the state of Utah[6] and in the US Congress,[7] that give PKI signatures standing under the law.

Despite a tremendous push from management, security professionals, consultants, and vendors, the market and the general public have been slow to adopt PKI. Explanations abound, including usability (PKI clients are harder to use than simple user names and passwords) and cost (users must purchase some certificates, and even free certificates have some deployment cost). Finally, some experts insist that the claims made for PKI are unjustified, because computer viruses and other kinds of malicious software can compromise private keys or make people think that they are signing one message when in fact they are signing another.

This technique uses an email address as a universal identifier and the ability to receive email at that address as a kind of authenticator.

Another reason for PKI's slow adoption is that its capabilities generally do not match typical user requirements. Identification and authorization are certainly two requirements for e-business, but I believe that PKI does not adequately satisfy other e-business requirements.

One requirement is *delegation*. Many professionals must delegate their authority. With a typical user name–password system, a professor can let her assistant read her email by sharing her password; to remove this delegated authority, the professor just changes her password.

PKI makes delegation much more difficult. To maintain legal assurances of privacy, authentication, and nonrepudiation, PKI systems require that individual PKI users never share their private keys with others. Instead, PKI-based systems require that elaborate delegation arrangements be codified and set up in advance—for example, professor Alice must explicitly authorize her assistant Bob to be able to read her mail and perhaps to file grade reports on her behalf if she is unavailable. In theory, such delegation could be done by registering Bob's identity with some third party or by having Alice issue Bob a certificate. But in practice, setting a system up for delegation requires in-depth planning on the part of application designers, programmers, and users. For this reason, many PKI systems do not allow for delegation.

While we certainly can map out such relationships in advance, the process is difficult and time-consuming. For that reason, even systems that rely on PKI for identification sometimes fall back on user name–password authentication because of its relative ease of delegation. For example, the US Armed Forces has deployed roughly 4 million client-side certificates. Nevertheless, according to Richard Hale, at the US Defense Information Systems Agency, many mission-critical Web sites—especially those used in combat situations—rely on user name–password authentication precisely because individuals can share user names and passwords without prior arrangement.

Email-based identification and authentication

While many organizations continue to invest in PKI, another technique for identifying and authenticating Internet users is rapidly emerging in the marketplace. This technique uses an email address as a universal identifier and the ability to receive email at that address as a kind of authenticator. Nearly every major Web service provider, including eBay, PayPal, Amazon, Yahoo, and Apple, among others, has deployed some form of EBIA.

Today, the primary use of EBIA is in systems that let users recover lost or forgotten Web passwords. Most online services let users register an email address during account creation; if a user forgets his or her password, the system automatically generates a new one and sends it to the registered email address. (Some systems do not bother to generate a new password, and simply email the old one.) Other EBIA systems facilitate password resets by sending users an HTML link; when users click on the link, their Web browser opens to a page that lets them create a new password.

Similarly, many Web sites now require people registering with them to use their primary email address as their user name. This practice overcomes a common but important problem for Web sites: namespace collisions. When users can pick their own user names, two or more users can choose the same one. (Although subsequent users must always pick names that are not in use, the profusion of multiple user names is itself a usability problem.) But because email addresses are necessarily unique and it's easy to verify ownership of an email address (by sending an HTML link that requires a response), using email addresses as user names avoids the possibility of conflict.

At first glance, EBIA might seem unsecure and, therefore, unwise. After all, the vast majority of Internet email travels without cryptographic protection: Someone or some thing could read or modify email without detection while the message is in transit. Indeed, several commercial systems do just that—Yahoo, for example, inserts advertisements into email messages and, perhaps more significantly, will alter email that appears to resemble JavaScript. What's more, key employees at many businesses and Internet service providers (ISPs) can browse or perform keyword searches on users' mailboxes. Given this lack of security, relying on email to prove identity or facilitate financial transactions seems unwise.

Indeed, many security professionals have criticized EBIA systems, complaining about the practice of emailing unencrypted passwords, their reliance on email addresses as identifiers, and on flawed implementations that can send "password resets" to any email address, rather than only to the address on file. Microsoft's .NET Passport service, for example, once let any individual who knew how to exploit such a security flaw in its password reset system seize any Passport account—although the flaw was corrected after it was publicly disclosed.[8]

But EBIA's widespread use in today's online world implies that these security risks are manageable, especially considering that the obvious alternatives to EBIA (for example, placing a phone call or sending a letter through the postal service) are prohibitively expensive for many Web operators.

Even if EBIA is not here to stay, it is here now. Rather than attacking the practice, a more productive approach for security professionals is to explore the reasons for its success, develop guidelines for using it securely, and create strategies for transitioning to more secure alternatives.

What EBIA gets right

EBIA has been successful because it combines ease of use with a limited challenge–response system that is not trivial to defeat. As with SSNs, people can "identify" themselves using a short and easy-to-remember character string. (Arguably, email addresses are even easier to remember than SSNs.) But like PKI, EBIA separates identification from authentication: like a name, the email address is an identifier, but authentication is based on the ability to receive email at that address.

A key advantage of EBIA over PKI is that PKI requires specialty software and a mutually trusted CA. EBIA, on the other hand, can work with any email client (or even with Web-based email), using email addresses available from hundreds of thousands of different email-granting organizations (ISPs, companies, schools, government organizations, and so on).

EBIA also matches business and personal requirements better than traditional PKI systems because

- PKI tends to establish a single—personal—identifier. Because it's difficult to obtain certificates, PKI encourages individuals to use a single certificate for many different applications. This makes it possible for Web sites and other online service providers to tie together different transactions, possibly resulting in an invasion of privacy. By contrast, email addresses are much easier to create and destroy. Individuals can choose whether they want linkable identifiers or different email addresses for different relationships.
- In some circumstances—for example, when corresponding with a bank regarding a specific account — individuals using EBIA must identify themselves, but in many cases, it is not necessary that an email address map back to an identifiable person. Although it is possible to create purpose-built PKI certificates that do not disclose a person's identity—or to use mathematical approaches that reveal only specific attributes[9]—in practice, most PKI systems have policies requiring client certificates to contain individuals' legal names.
- The EBIA process demonstrates to users and implementers that perfect identification is impossible. Someone can gimmick or bypass every system devised to identify one human from another: Unscrupulous people can forge passports, steal SSNs and private keys, and tamper with biometric databases. Because PKI gives the illusion of a mathematically perfect and unchallengeable identification, organizations are typically less prepared for cases in which PKI identification fails (for example, because of software flaws, stolen keys, or improperly granted certificates). Because EBIA is a weaker form of identification than PKI, organizations that rely on it have strong incentives to create additional security measures (for example, increased auditing, profiling, and fraud detection). Redundancy, resiliency, and provisions for handling an occasional error can create a unified EBIA system with better privacy and security guarantees than off-the-shelf PKI technology.
- With SSNs and PKI, someone can use a stolen private key against its rightful owner again and again, without the owner ever finding out. By contrast, EBIA is self-auditing. When an email addresses is used for verification, an address owner discovers that the verification is taking place because of the confirmatory email messages (assuming the owner actively uses the email account). If the individual cannot receive the confirmatory email message because someone changed the account password, there's a high probability that the individual would contact the email provider to have the password reset. If that user continues to have no control of the email account, he or she will realize that something was wrong and, presumably, investigate the cause.

  Of course, EBIA only is self-auditing if the email address owner remains the same and—even then—only if the owner frequently checks the email account.
- EBIA enables a competitive market for identity and authentication services. Different email providers have different standards for security. Some organizations require extensive proof before resetting a password; oth-

Another reason for PKI's slow adoption is that its capabilities generally do not match typical user requirements.

ers will do it over the phone. Some individuals are so untrusting that they insist on running their own email service, but most outsource their email to trusted providers. An attraction of EBIA is that individuals can obtain their email addresses from organizations that make them feel comfortable.

Moreover, when large providers do have poor security practices, this information spreads rapidly. Because email addresses contain the provider's trademark after the @ symbol, even relatively unsophisticated organizations and individuals can learn about the provider's problems and make informed decisions about moving their email elsewhere. Such transparency generally is not the case with PKI approaches: even though PKI explicitly bases a certificate's trustworthiness on the organization that issued it, many programs that implement PKI (such as Netscape Navigator and Internet Explorer) do not make the granting CA immediately visible.

- Most importantly, EBIA's trustworthiness emerges organically as the result of other interactions between email participants. PKI, by contrast, is a top-down identification regime created solely for that purpose.

Identification for roles and groups

A *role* is an identity that is typically created for a particular task, rather than for use by a particular individual. A *group* is an identity that is shared by many individuals. Whereas roles and groups are typically difficult to implement with PKI, they are easy to implement with EBIA.

Like delegation, implementing roles and groups with PKI can be difficult. As a result, many organizations deploying PKI can be tempted to use individual certificates for applications that should be role-based. For example, instead of creating a certificate for the accounts payable department, a vendor Web site might rely on certificates issued to specific individuals in accounting. This can cause problems when the individual leaves the department.

But with EBIA, roles and groups easily can be arranged by creating email addresses that route to multiple individuals. For example, the email address stopit@mit.edu goes to a team of people which deals with abuse and harassment issues.

You could criticize EBIA by saying that there is no way to determine whether a singular person or a group of individuals use an email address. In fact, this ambiguity is one of EBIA's advantages: email addresses for roles or groups can be trivially established and used instead of personal email addresses without having to set up a lot technology. Role-based email addresses are simply email addresses with multiple recipients. In fact, the outside supplier doesn't even need to know that email sent to the address is not going to a personal email address but, instead, to several individuals.

Related work

Identity-Based Encryption[1] (IBE) is a system that uses an email address to create a public and private pair of keys for traditional encrypted email. The public key is created using a recipient's email address and public system parameters. The private key is created using the recipient's email address and both public and private system parameters. Thus, any user can create a public key for either participants or nonparticipants in the identity-based system. Participants can decrypt these messages as soon as they receive them; nonparticipants easily can become participants by obtaining their private key from a system coordinator. IBE is different from email-based identification and authentication in that IBE is encryption-based, but an organization could use email-based identification and authentication (EBIA) to distribute IBE-created private keys.

Authentify (www.authentify.com), a small Chicago-based firm, has developed a system to authenticate users based on their ability to receive a telephone call at a pre-designated telephone number. Designed for the financial service industry, users go to a Web site, enter their account information, and are then called by Authentify's computers. When the user picks up the telephone, he or she is prompted to enter a PIN displayed on the Web page.

PGP Inc.'s (www.pgp.com) PGP 8.0 Universal security appliance can be configured to automatically create a public/private key pair for recipients of email messages not registered with the system. Instead of receiving an encrypted email, the users are sent a link that can be used to access the appliance's built-in Web mail server. The sender can further protect the message by creating a passphrase.

RSA Security (www.rsasecurity.com/products/mobile/) and others (Including Min Wu et al. at MIT[2]) describe schemes for authentication at Web sites based on the ability to receive short message service (SMS) messages on mobile phones. The RSA scheme involves sending a one-time password to a mobile phone, which the recipient then types into a Web browser; Wu's scheme uses SMS and Wireless Application Protocol (WAP) on the cell phone to confirm a session that takes place on a conventional Web browser.

References

1. D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *Lecture Notes in Computer Science*, vol. 2139, 2001, Springer-Verlag, pp. 213–229.
2. M. Wu, S. Garfinkel, and R. Miller, "Secure Web Authentication with Mobile Phones," Student Oxygen Workshop, MIT Computer Science and Artificial Intelligence Laboratory, 2003; www.simson.net/ref/2003_wu_sow.pdf.

Establishing best practices for EBIA

EBIA is a powerful technique, and its substantial use throughout the market demonstrates that it is a workable solution. But this is damning with faint praise: SSNs also are a powerful technique with broad market acceptance, yet their simultaneous use as identifiers and authenticators aggravates identity theft. The fact that a technique is widely used is not proof that the technique is secure. Risk must be managed.

EBIA's use surely will increase in the coming years; however, users and providers must understand its limitations:

- EBIA security depends on the security of email servers and passwords. Today, most email travels over the Internet and is stored without encryption; a large number of Internet users download their email using the Post Office Protocol (POP) and unencrypted passwords.
- Email content is accessible to server operators. Without encryption, system managers can intercept, read, and make copies of email messages destined for end users.
- Different individuals at different times might use the same account. Some email providers lock a user name when the account terminates. Others do not. At some companies, a departed employee's email might forward to the person's replacement. For these reasons, businesses sending trusted information should not assume that the owner of an email address today will be the same one who owns the address tomorrow.
- There is no reliable way to match email addresses to legal names. One of PKI's great promises was that people would be able to reliably identify themselves online with their legal names. Some visionaries speculated that using certificates with additional fields, such as "age" or "sex," would enable the creation of pornographic Web sites inaccessible to minors, or "women only" Web sites inaccessible to men. EBIA makes no such assurances—although it could, if email providers were willing to offer some sort of authentication service. (In fact, suitably motivated email providers could leverage EBIA into a full-fledged PKI by providing certificates for their users if they requested them.)
- Being able to receive email sent to an address within a domain does not imply a recipient's affiliation with the organization that owns the domain. It is trivial to forward email from one computer to another. Just because a person can receive email at an MIT email address, does not imply that person's affiliation with MIT.
- Spam filtering can block authentication attempts. With an increasing number of users employing spam filtering, there is a growing chance that EBIA messages might not make it past filters or might themselves elicit challenges from spam-filtering challenge–response systems. Users signing up for antispam systems should make sure that their providers do not filter out EBIA.

By understanding these limitations, we can establish a set of best practices for the continued use of EBIA.

For email service providers:

- When possible, stored email should be encrypted. Current open-source email systems (for example, sendmail, qmail, and postfix) store email without encryption, making it available to anyone with administrative access to a mail server. Even simple symmetric encryption for stored mail would significantly increase security.
- Never send email passwords over unencrypted connections. Email POP and Internet Message Access Protocol (IMAP) servers frequently accept clear-text passwords from unencrypted email connections. Configure them to use encrypted connections (for example, POP over SSL) or challenge–response authentication mechanisms (Authenticated Post Office Protocol—APOP or IMAP with CRAM-MD5) that do not rely on clear-text password exchange.

For sites that use EBIA:

- Never base authentication on the ability to send email from an address; base it on the ability to receive email at an address. All email clients let users specify which "from:" address to use. Moreover, many people receive email at one address and send it from another one. Authentication should be based solely on the ability to click on an emailed link, or to reply to an email message in such a way that preserves a code word or nonce in the subject line. (Failure to follow this dictate has caused persistent problems for someone trying to unsubscribe from mailing lists; some mailing list programs expect unsubscribe requests to originate at the email address being unsubscribed, rather than simply embedding a clickable link in a message.)
- Because there is no way of knowing whether an email address still belongs to the original holder, organizations that rely on EBIA should not send out authentication messages unprompted. Instead, users desiring authentication should be the ones to initiate EBIA—and if the provided email address matches the address on file.
- Because the same password can be used at several organizations, a service that employs EBIA to reset a password should never send the user's old password to the registered email address; instead, the service should create a new password and send that one instead.
- Because they are themselves security-related messages for which authenticity and integrity are important, sites that initiate EBIA messages should digitally sign them. Once such practice is commonplace, email clients automatically could recognize digitally signed EBIA messages as being legitimate, thus avoiding antispam filter-

ing. Client-side software also could differentiate between legitimate EBIA mail and attacker-sent spoof mail. (PayPal, for example, has had persistent problems with such spoof mail sent to its customers;[10] the company now asserts that customers are responsible for discriminating between legitimate email messages and fakes.) Sadly, both the Open Pretty Good Privacy[11] and the Secure/Multipurpose Internet Mail Extensions (S/MIME)[12] standards pose significant usability problems for users who receive email messages signed with a standard that is not implemented by their browser: such signatures appear as indecipherable attachments. (In the case of OpenPGP, the signed message might appear as a blank message with two attachments!) That is, both of these formats impose usability burdens on non-users. I am presently developing new formats for signed email that do not have this problem. (More details can be found at http://stream.simson.net/.)

- Organizations relying on EBIA must provide mechanisms for users to change their registered email addresses without invoking EBIA. A user might lose the ability to receive email at a certain address before he or she has an opportunity to update it. Thus, online systems that use EBIA should have alternative authentication mechanisms, such as passwords, to update a registered email address. These authentication mechanisms work hand in hand with EBIA. (For example, services like eBay and PayPal make users choose passwords and register email addresses.) When a password is lost, EBIA performs a password reset. Alternatively, when an email address is lost, the password can be used to register a new one. When both are lost, human contact is necessitated.

Organizations relying on EBIA can use it as a steppingstone to PKI by associating a public key with each user's email address. Instead of making an all-or-nothing jump into the world of PKI, organizations could start by simply storing an optional public key for each email address that they have on file. These public keys could be used as a second level of verification for incoming email messages: matching signatures would let an organization trust that successive email messages from the same address actually came from the same person (or at least the same e-mail client). This strategy does not require the use of a third-party CA because the organization is simply interested in matching public keys with email addresses, not in using the public keys to determine an individual's legal identity. By following these rules, organizations can use EBIA for a variety of tasks, including account recovery, password resets, confirmation of high-value transactions, and eventually bootstrapping Internet users to more secure authentication mechanisms.

In the absence of a universal PKI technology deployment, we're increasingly using email addresses as identifiers, and the ability to receive email sent to an address as an authenticator. Instead of fighting this trend, security practitioners need to understand it and develop techniques for using EBIA effectively and securely. EBIA is here today and, at least for the foreseeable future, here to stay. Individuals and companies interested in deploying PKI should work on ways of integrating PKI with EBIA. □

Acknowledgments

Inspiration for this article originated at Jean Camp's 2003 Conference on Identity (www.ksg.harvard.edu/digitalcenter/conference) at the Harvard Kennedy School of Government. I thank Dave Clark and Rob Miller at MIT and Beth Rosenberg for reviewing earlier drafts of this article and offering useful comments and suggestions. Joe Pato at HP provided useful references. I also thank IEEE Security & Privacy*'s anonymous reviewers for their valuable comments.*

References

1. R.E. Smith, *Social Security Numbers: Uses and Abuses*, *Privacy J.*, 2002; www.privacyjournal.net.
2. Synovate, *Federal Trade Commission—Identity Theft Survey Report*, Sept. 2003; www.ftc.gov/os/2003/09/synovatereport.pdf.
3. S. Garfinkel, "Risks of Social Security Numbers," *Comm. ACM*, vol. 38, no. 10, 1995, p. 146.
4. T. Dierks, "The TLS Protocol, Version 1.0," RFC 2246, Network Working Group, Jan. 1999.
5. "Digital IDs: The New Advantage," VeriSign, 1999; www.verisign.com/repository/clientauth/clientauth.html.
6. Utah Digital Signature Act, Utah Code §§ 46-3-101 to 46-3-104, 1996; http://cio.utah.gov/initiatives/digitalsignatures.htm.
7. Electronic Signatures in Global and National Commerce Act (ESIGN), 101(c)(1)(C)(ii), US Congress, 2000.
8. "Security Issue in Microsoft .NET Passport Is Resolved," Microsoft, May 2003; www.microsoft.com/security/passport_issue.asp.
9. S.A. Brands, *Rethinking Public Key Infrastructure and Digital Certificates: Building in Privacy*, MIT Press, 2000.
10. A. Gilbert, "Email Scam Tries to Fool PayPal Users," CNet News.com, 7 Mar. 2003; http://news.com.com/2100-1018-991639.html.
11. M. Elkins et al., "MIME Security with OpenPGP," RFC 3156, Network Working Group, Aug. 2001.
12. S. Dusse et al., "S/MIME Version 2 Message Specification," RFC 2311, Network Working Group, Mar. 1998.

Simson L. Garfinkel *is a doctoral candidate at MIT's Computer Science and Artificial Intelligence Laboratory and a commentator on information technology. His research interests include computer security, secure systems usability, and information policy. He is the author of* Database Nation: the Death of Privacy in the 21st Century *(O'Reilly, 2000) and coauthor of* Practical UNIX and Internet Security *(O'Reilly, 2003). Contact him at simsong@mit.edu; http://simson.net.*