EGGS IN BASKETS:
DISTRIBUTING THE RISKS OF ELECTRONIC SIGNATURES

By Benjamin Wright

Summary: The risks with electronic signatures might be addressed with any number of alternative strategies. One strategy, designed to appeal to the common person, would use biometrics to spread the risks so that no particular feature of the signing process, such as a private key, is highly important.

Electronic commerce brings questions about how to sign, or evidence legal approval of, electronic documents. Evidence that a person approved a particular electronic document might be gathered many different ways. The article evaluates two ways, one using public- key cryptography and the other using pen biometrics.<1>

The signing of a document is a social event, not a scientific event. It is an act in which an individual -- Alex -- evinces approval of the document so that someone else -- Bob -- can perceive that approval, understand it, and later prove it to other people. But the bonding of Alex to the document is never a perfectly reliable process. Whatever evidence exists to support the bond is subject to challenge. In other words, signing documents involves risk.

TRADITIONAL INK AND PAPER STRATEGY

Many risks afflict the traditional signing of a paper document. First, there is no standard method for signing in ink. Alex is not taught or required to sign documents in a forensically reliable way. Alex is free to sign in any way he chooses, and to change his signature from minute to minute. For any given signing, Alex is free if he so desires to use as his signature any strange and indecipherable scribble. Whether any given document signed by Alex does or does not contain Alex's usual, verifiable signature is for all practical purposes a secret. Rarely is Alex's signature compared against specimens to confirm authenticity.

Ink signatures can be forged. There is no guarantee that any given ink signature can be verified by forensic science. Science can only offer an educated opinion as to whether the signature is authentic, and it can do so only under the right circumstances (including, for example, the availability of several good specimen signa-

tures).

Other risks impede the linking of Alex to a given paper document. If the document is multiple pages in length, one or two of the pages could be switched after the document was signed. There is even the risk that the document is organized in an ambiguous or confusing way, so that an observer cannot discern for certain which parts of the document Alex agreed to and which parts he did not.

These risks mean that in the event of a dispute,<2> it is not always easy to tie Alex to specific words in a paper document. When Alex signs a document and gives it to Bob, Bob is not guaranteed that he will later be able to prove that Alex signed it.

Alex might raise any number of objections to repudiate the document. Conversely, Alex is not guaranteed that he can repudiate a document that he in fact did not sign.

Under American law, the burden of proving that Alex did sign is normally on Bob. This burden motivates Bob, at the outset of the transaction, to seek evidence of Alex's responsibility from things other than simply Alex's signature. This may mean that Bob would ask Alex to acknowledge his signature before a notary. More commonly, it means that Bob establishes a relationship with Alex in which they exchange feedback between each other -- Bob asks for partial advance payment, Bob sends acknowledgments to Alex's independently verified address, or the like. The feedback reduces the risks to Bob.<3>

The myriad risks with a paper and ink signing are distributed across a number of features of the signing ritual -- Alex's style of signing, Alex's secret choice whether to use his usual signature, the content of the signed document, the facts external to the document (such as any interaction between Alex and Bob) that place it in a historical context, the competence of the person who opines on the authenticity of the signature, and so on. In other words, the eggs are spread into many baskets. No single egg is highly reliable or highly important.

In a dispute over the authenticity of a document, the fact finder (the jury) does not look at the signature in a vacuum. Rather it considers all the relevant facts and circumstances -- the historical context of the document and all the ambient clues (such as corroborating records or testimony) that might bear on the authenticity question.

Just as risks plague the authentication of paper documents, so they will plague the authentication of electronic documents. To expect perfect binding of an individual like Alex to the words of an electronic document is not realistic.

To bind Alex to his electronic words, inventors might craft any number of strategies. One that has attracted attention is embodied in the Digital Signature Act adopted by the Utah Legislature in 1995 (Utah Act), which is codified at Utah Code Annotated Section 46-3-101 et seq. It contemplates that public-key cryptography would be used in the context of a large, global network of government-licensed certification authorities (CAs).

PUBLIC-KEY CRYPTOGRAPHY DESCRIBED

Public-key cryptography provides a mathematical scheme for arranging computer data (e.g., an electronic expense voucher or medical record) such that its integrity and origin can be proven.<4> Public-key cryptography involves the use of two keys (special strings of data), a private key and a public key, which are assigned to a user (Alex). Each key bears a complex mathematical relationship to the other. As the name suggests, the private key is intended to be kept secret, so that only Alex can access it. The public key, however, is not intended to be kept secret. It can be published so that outsiders can know it.

Suppose, for example, that Alex wants to sign a document for Bob in a way that confirms the document was really signed by Alex. After viewing the document's content, Alex would use his private key and a crypto program to attach a "digital signature" to the document. The digital signature is a short unit of data that bears a mathematical relationship to the data in the document's content. Bob then could confirm the document's authenticity by using Alex's public key and a crypto program. If the document was not altered between the time Alex signed it and the time Bob confirmed, then the program informs Bob that the document was signed by someone possessing Alex's private key. Bob might infer then that Alex did sign the document. If the program cannot inform that the document was signed with Alex's private key, then Bob infers either that the document was not signed with Alex's private key or that it was changed after it was signed.

Public-key cryptography can be used in endlessly creative ways. The Utah Act contemplates using it in one particular way, what this article calls the "Utah Strategy." (The Utah Act is an admirable intellectual work; within limits it may support other strategies.)

THE UTAH STRATEGY

One of the risks in signing with public-key cryptography is that the person using the public/private key pair might not be the right one. A person claiming to be Alex may in fact not be Alex. To alleviate this risk, the Utah Strategy imposes an elaborate scheme for binding Alex to a particular public/private key pair.

First, a CA would be licensed by state government to ascertain the identities of people like Alex and link them to their key pairs. When the CA confirms Alex's identity and his control of the requisite key pair, it must ask him to use a new "distinguishing name," a computer code that labels Alex uniquely in all the world, so that the nominal link between Alex and his key pair is unmistakable. After the CA confirms Alex's identity, distinguishing name, and key pair, it issues a certificate affirming that Alex's distinguishing name is associated with the public half of the key pair. (The certificate must expire within three years, which implies that Alex must re-register with the CA regularly.) Alex is then obligated by the Utah Act not to allow his private key to fall into the hands of someone else. If Alex neglects his obligation, it would be difficult for him to avoid responsibility for documents signed with the key (e.g., demands for the withdrawal of money from his bank account), even if he did not approve them.

The Utah Strategy entails the keeping of secrets. Alex must keep his private key secret. But Alex will not be able to remember his key, so he will have to store it on a computer device such as a smart card and then keep the device in a safe place. In turn, the vendors of the smart card and its various components will need to keep secrets. If the vendors fully disclose the methods they use to control the private key, then it will be easier for an attacker to steal the key.

Strictly speaking, public-key cryptography does not reduce risk in the signing of an electronic document. It transfers risk. It can be very effective in showing whether a particular document was signed with a certain private key.

But this transfer of risk does not necessarily result in the elimination or even the reduction of risk. Risk simply shifts onto the private key. That key becomes the object of any criminals who want to cheat Alex or Bob. They will try tricking Alex into revealing the key or temporarily surrendering control of it. They will endeavor to compromise the software that controls the key and its functions. Or they will steal and unlock the device, such as a smart card, in which Alex stores

his key. If under the Utah Strategy millions of people were issued smart cards containing valuable private keys, then an underground industry of criminals would devote itself to corrupting those cards and the infrastructure that underpins them.

Not only does the Utah Strategy shift risk to the private key, it concentrates the risk there. The Utah Act gives recipients like Bob strong reason to expect that if a document is signed with Alex's private key then Alex is legally responsible for the document. Utah Act Section 46-3-401 provides that a document signed with a digital signature is normally presumed to be signed by the person owning the relevant private key (so long as his public key is certified by a licensed CA). This presumption reduces Bob's incentive to gather or consider any evidence other than the digital signature when he evaluates whether he can prove that Alex is responsible for a document. It allows Bob to forego the trouble of establishing a relationship with Alex or exchanging acknowledgements or other feedback with him to confirm his responsibility.

This presumption in turn gives Alex powerful incentive to protect the key. The incentive is much greater than the incentive consumers have to protect their automated teller machine cards. (Under the Electronic Funds Transfer Act, 15 U.S.C. 1693-1693r, consumer liability for a stolen ATM card is often limited to just $50.)

Under the Utah Strategy, control of Alex's private key becomes all important. In other words, virtually all the eggs are placed in one basket -- the private key (and the technology such as smart cards that protect that key from villains). This allocation of risk may make sense for some transactions, particularly high-end financial deals initiated by sophisticated people. But a common person like Alex may not feel comfortable with it. He may not like
1. dealing with a bureaucratic CA,
2. associating himself with a computerized distinguishing name, or 3. having responsibility for an extremely powerful private key.

Fortunately, the Utah Strategy is not the only way to allocate risk in the signing of electronic documents. Alternative strategies would spread the eggs among many baskets.

PENOP

One strategy for spreading risk would use PenOp(tm).

PENOP DESCRIBED

PenOp employs a pen biometric technology. It is a computer software component that augments the function of other computer applications -- such as applications that control electronic documents. PenOp has two primary features:

1. The Signature Capture Service *(SCS)* captures and permits the storage of certain data associated with the manual inscription of a signature (autograph) on the screen of a pen-based computer (or a digitizer pad on a PC). The SCS must work with a "Client Application," which is software that informs the pen computer user what he is doing and prompts him when and how to do it. A Client Application can be designed to manage an electronic document such as an expense voucher.

In coordination with the Client Application, the SCS receives information, such as a user ID or a name, showing who the user (Alex) claims to be. It then prompts Alex to inscribe his signature, using a stylus (or pen), to a window on the computer's screen. It supplies the wording of the prompt in the window, known as the "Gravity Prompt," which indicates the purpose for which the signature is being captured. The Gravity Prompt normally refers toan electronic document that is accessible to Alex through the pen computer.

As Alex moves the stylus across the screen, an image appears that traces the movement of the stylus. Thus he sees his autograph. At the same time, the SCS measures certain features of the inscription event, including the size, shape, and relative positioning of the curves, loops, lines, dots, crosses and other features of the signature being inscribed, as well as the relative speed at which each feature is imparted. The results of these measurements are known as "act-of-signing statistics." Alex then has the option, by tapping indicated buttons on the screen, of approving the inscription event, retrying it, or aborting it.

If Alex taps the approval button, the SCS calculates a checksum, or a brief string of data, that represents the content of the electronic document referred to by the Gravity Prompt. This checksum is not a complete statement of the original document, and the original document cannot be derived from the checksum. But the checksum bears a mathematical relationship to the document. If the document is changed, then it can no longer be mathematically matched with the checksum. (PenOp creates checksums using the MD5 digest algorithm by RSA Data Security, Inc.)

Next, the SCS compiles the following data (the "itemized data") and computes a second checksum from it:

* the first checksum
* the act-of-signing statistics
* the date and time of signing (as represented by the computer operating system under which the SCS is operating)
* the identity of the particular machine on which the signing occurred (based on identity information programmed earlier in the SCS)
* the claimed ID of the user (Alex)
* the words that appeared in the Gravity Prompt
* (optionally) data reflecting the graphic image of user's signature

The SCS creates the second checksum in two steps. First, the SCS retrieves a secret key previously programmed into the Client Application and uses that key to encrypt the itemized data. (This is the "first level of encryption.") Second, the SCS calculates from that encrypted data the second checksum. The second checksum establishes a link between the itemized data and the Client Application.

Finally the SCS encrypts the itemized data, plus the second checksum, using a different algorithm, one which does not use a secret key from the Client Application. This is the "second level of encryption". The resulting encrypted string of data -- called the "Biometric Token" -- is a tamper-resistant representation of the event in which Alex inscribed his autograph.

2. The Signature Verification Service *(SVS)* reports the probability that a particular signature is authentic. First, in authorized enrollment sessions, the SCS captures and the SVS holds, in a database, act-of-signing statistics for a user like Alex who has been identified to the SVS.

Later, the SVS may be presented with a particular Biometric Token and directed to evaluate whether it is a product of an authentic inscription of the signature belonging to the user identified in the token. The service decrypts the token and then compares the information therein with the signature statistics stored earlier in its database. Based on this comparison, it issues a "signature match percentage", e.g., 50 percent or 72 percent, and reports this percentage to a Client Application (software configured to make use of the report). The SVS applies scientific principles deemed relevant by PenOp's developers.

PENOP STRATEGY

PenOp might be used to "sign" electronic documents such as contracts, expense reports, or medical records. Here is one strategy for doing so, what this article will call the "PenOp Strategy." Under this strategy, Bob seeks merely to have Alex attach a Biometric Token to an electronic document for the purpose of "signing" it. Bob does not seek to verify the Biometric Token at the time he receives it, just as he would not verify Alex's autograph at the time he receives paper from Alex.

To start, a Client Application within a pen computer is configured to display to Alex the data within the document in question (text, graphics, and so on, all in digital format). The Client Application then calls the SCS to write a Gravity Prompt, inviting Alex to "sign" the document by inscribing his signature within a window on the computer screen. The SCS also presents Alex a button for approving the inscription. If he inscribes and approves, SCS captures the necessary data and creates a Biometric Token. The SCS delivers the token to the Client Application for storage in a way that identifies the token as being related to the signed document.

If, at a later date, a third party such as a court wishes to verify that Alex did "sign" the document, it could obtain the PenOp SVS, introduce Alex to it, and use it with the help of an expert to verify (to the degree possible) that the Biometric Token represents an inscription by Alex.<5> A test could also be made (using the checksums in the Biometric Token) to establish whether the document to which the token is linked is the exact document used at the time of the token's creation. Another test could be made to confirm that the token was made with the identified Client Application.

The PenOp Strategy is similar to the traditional paper and ink strategy. Direct signature verification occurs only on rare occasions, and the full burden of proving that Alex signed a document rests with Bob. Bob is therefore motivated not to rely greatly on the signature; he will want to get evidence and security from other sources, such as advance payment or feedback from Alex.

RISK ALLOCATION WITH PENOP

Like the traditional paper and ink strategy, the PenOp Strategy allocates risk

across multiple factors (spreads the eggs to many baskets).

The creation of a biometric token that falsely appears to come from Alex requires the attacker to defeat security features (break eggs) that are supplied by these three different people:
1. The developer of the Client Application
2. The developer of PenOp
3. Alex

1. A Biometric Token must be made with the aid of an identifiable Client Application. The Client Application supplies the secret key that is used in the "first level of encryption." If an attacker stole many proprietary secrets from PenOp's developer he could learn in the abstract how to create false biometric tokens (see below).<6> But to create a false token that convincingly links a specific transaction to Alex, the attacker would also need to steal the secret key and other information from the developer of the Client Application. The degree and character of security that the Client Application employs is to be decided by its developer. That developer may, if it chooses, employ multiple secrets that are spread among multiple segregated parties (thus distributing more eggs into more baskets). It can employ audit trails, secure timestamps, physical access barriers, and even public-key cryptography as parts of its security (the Client's secret key might be the private half of a public/private key pair). The greater the security, then the greater the forensic value of documents signed through that Client Application.

Before Bob relies very much on a Biometric Token, he will want to know about the reliability of the Client Application that helped create it. (Bob would have a similar concern if he were relying on a document signed by Alex with private key cryptography. The reliability of the link between a public key and a document signed by that key depends on the reliability of the software that controls the key and exposes it to the document it signs.)

2. The PenOp software employs a complex array of secrets that are difficult for an attacker to obtain, understand, and use. First, the methods PenOp uses to measure and record an act of signing are known only to PenOp's developer, and those methods can change over time. Second, for the "second level of encryption" PenOp uses a novel encryption method that achieves these goals:

* Neither the SCS nor the SVS software possesses the key that encrypts a token at the second level of encryption.

* Neither the developer of PenOp nor the developer of the Client Application possesses the key that is used for the second level of encryption.

* Someone possessing the object code to the PenOp SVS software could decrypt a biometric token (at the second level of encryption) for purposes of a preliminary analysis of the checksums (linking the token to the original document) and the signature statistics (linking the token to Alex), but that person would not have the information to falsify tokens or inconspicuously corrupt them. A deeper analysis of the signature statistics would require expert advice and access to secrets kept by PenOp's developer.

* To understand and replicate the second level of encryption, an attacker would need to steal the source code to the SCS and the SVS. The developer of PenOp intends to keep the source code a secret.

* A very sophisticated and determined attacker might be able break the second level of encryption. But the breaking of that level allows the attacker to overcome just one hurdle (break just one egg) in his fraudulent effort to create a signed document. What is more, the second level of encryption is highly resistant to known plaintext attacks.<7>

3. To create a convincing biometric token purporting to be from Alex, an attacker would need to obtain extensive information about Alex's signing behavior. For the attacker, this would be a considerable burden.

Even if an attacker successfully tricks the developer of a responsible Client Application, the developer of PenOp, and Alex into disclosing the necessary information, the attacker would still have much work ahead of him (more eggs to break). To perpetrate a fraud, he would have to fabricate a convincing transaction, one that makes sense under the prevailing facts and circumstances. It would have to be consistent with the types of transactions Alex would have entered at the time, including the records that Alex himself and other interested parties would keep. If, for example, Alex were a school teacher of modest means, a corporate debenture that appears to bear his signature would not be convincing.<8> Bob would have a hard time carrying his burden of proof that Alex signed that document.

AUTHENTICATION IS AN ENDLESS JOURNEY

No particular application of paper and ink, the Utah Strategy, or the PenOp Strategy can provide a perfect bond between Alex and the words of a document. The development and use of authentication technology is a dynamic process. It is not a destination; it is an endless journey in which the good people hurry to stay a step or two ahead of the bad people. The paper and ink tools that provided adequate authentication in the year 1900 do not necessarily provide the same level of authentication in the year 1995 (consider how the sophistication of money counterfeiters has grown in the past 95 years). The computer security tools that provided adequate authentication in 1970 do not provide the same level of authentication in 1995.

Similarly, the tools needed to provide adequate protection for a private crypto key in 1995 will be different from those needed to provide equivalent protection in 2010. And, yes, the tools needed to protect a PenOp application from abuse in 1995 will be different from those needed in 2010. As PenOp becomes more popular, PenOp's developer may need to divide and spread PenOp's secrets (some of its eggs) across a larger number of people. Under both the Utah Strategy and the PenOp Strategy, systems developers will have to work endlessly to keep their secrets out of the hands of criminals.

In principle, neither the Utah nor the PenOp Strategy is an inherently superior method for connecting Alex to his electronic words. Each is an approach for staying one step ahead of the bad people. Whether any particular application of these strategies is adequate will depend on all the facts and circumstances of a particular transaction.

The chief difference between the strategies lies in the ways that people will use them. The Utah Strategy emphasizes the investment of many eggs in one basket -- the private key; whereas the PenOp Strategy (like the old paper and ink strategy) emphasizes the spreading of eggs across many baskets.

POPULAR ELECTRONIC COMMERCE

What does this difference in risk spreading mean? It means that the Utah Strategy may not appeal so much to members of the general public.

The Utah Strategy stresses the responsibility of Alex to protect his private key, and places a light burden of proof on Bob. In contrast, the PenOp Strategy, like

the traditional paper strategy, stresses that Alex and Bob act reasonably under the circumstances. Evidence of signing comes from all the relevant facts, with the full burden of proof being on Bob.

The Utah Strategy relies on the creation of a complex network of CAs, and requires Alex to register his identity with a CA (repeating the registration every three years) and to take a new distinguishing name (a socially and politically sensitive requirement). The PenOp Strategy, on the other hand, requires no advance planning or action on Alex's part. It caters to Alex's interests, and his instincts for how authentication should work, thus making electronic commerce attractive to consumers and common people.

==========

ENDNOTES

<1> This article does not consider whether any given method will be considered in law as a signature. For more on that topic, see Benjamin Wright, _The Law of Electronic Commerce_.

<2> In practice, disputes over the authenticity of commercial documents are rare. Of the many billions of commercial documents created every year, the authenticity of only a tiny fraction of the total is seriously contested in court. Among the reasons for this are that most people are happy with their commercial transactions most of the time and the facts and circumstances surrounding the documents (including the signatures, but also including the context and content of the documents) tend to show their origin and authenticity.

<3> See "Legal Identity and Signatures on the Information Highway," a component to Benjamin Wright, _The Law of Electronic Commerce_.

<4> Public-key cryptography is but one of many tools in the world of cryptography. Even within the field of public-key cryptography, there are many specific technologies. But for understanding the issues discussed here, it is adequate just to consider public key as a single, generic technology. Depending on how implemented, various public key technologies can perform some or all of the functions described here. Cryptography is a complex topic. Any reader who seeks to achieve certain results from cryptography should consult a competent professional. For more on cryptography, see Bruce Schneier, _Applied Cryptography_ (John Wiley and Sons) 1994.

<5> Alex's cooperation, although helpful, would not necessarily be required at the time his signature is verified. According to PenOp's developer, even if Alex is dead or refuses to cooperate, a limited forensic comparison could still be made between his signature, as captured earlier by PenOp, and one or more specimens of this signature as written on sundry paper documents such as checks, letters, contracts, or credit card receipts.

<6> As the PenOp Strategy is implemented, secret information will need to be divided and spread among segregated parties. Similarly, as the Utah Strategy is implemented, secret information about the private keys and their security (e.g., information about the function and control of smart cards and supporting software) will have to be divided and spread among segregated parties.

<7> A "known plaintext attack" is one in which the attacker treats the encryption algorithm as a black box. He takes a piece of known plaintext (e.g., a string of zeros), selects a key, and tries to discern how the algorithm works by running the text and key through the algorithm. He then analyzes the resulting encrypted text. For example, when a string of zeros is run with the key "KEY" through a very simple algorithm, the encrypted result might be "KEYKEYKEY".

But PenOp's second level of encryption is highly resistant to a known plaintext attack because it never encrypts the same block of data twice in the same way. The reason is that it uses a random encryption key.

<8> As the Information Age progresses, records about transactions become far more extensive and detailed than was true before. And the records become spread among many (and sometimes unexpected) parties, including sundry network service providers. The massive audit trail that will build up around commerce will make for an environment in which fraud is more difficult, not less. More of the facts and circumstances that surround a transaction will be recorded by independent third parties and other reliable means.

==========

June 22, 1995

Benjamin Wright (73457.2362@compuserve.com), a Dallas-based attorney, is special counsel to PenOp, Inc. (www.penop.com; tel: (800) 286-4137). He is also author of _The Law of Electronic Commerce: EDI, Fax and E-mail_, published by Little, Brown and Company (tel: (800) 331-1664; fax: +1-617-890-0875).

Mr. Wright is not an engineer, a computer scientist, or a forensics expert. He has

assumed the technologies identified here function and are implemented in competent, reliable, and credible ways and adequate records are maintained. No warranty is given as to the accuracy or completeness of the information in this article. The transaction of commerce is inherently risky, and nothing published by Wright advises which level of risk is appropriate for you.

For more articles on the legality of electronic commerce, see http://infohaus.com/access/by-seller/Benjamin_Wright.

Lessig
Taubman 210
79 JFK Street
Cambridge, MA 02138
617-495-1957