Message Authentication by Integrity with Public Corroboration

P.C. van Oorschot
Digital Security Group
School of Computer Science
Carleton University, Canada

ABSTRACT

One of the best-known security paradigms is to use authentication as the basis for access control decisions. We turn this around, and instead rely on access control (or more precisely, integrity) as the basis for authentication. We propose a simple, practical means by which data origin assurances for message authentication are based on corroboration, for example by cross-checking with information made available by a known source or at a specified location (e.g., web page). The security relies on the integrity of this corroborating information, and thus on access control on the hosting (or publishing) of this information. We do not explicitly require cryptographic keys for the corroboration step, or for the protection of corroborating information (e.g., it may be publicly posted), and thus our paradigm allows message authentication without direct dependence on private or secret keys. It may be characterized as *security by integrity.* Message authentication applications we discuss include email source authentication, and data origin authentication for digital signatures. Our work thus has application to problems including spam and phishing (e.g., where email with spoofed source addressing is involved), and addresses theft, extraction, or other illicit determination of digital signature private keys.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*authentication*; C.2.0 [**Computer Communication Networks**]: Security and Protection

General Terms

Security

Keywords

message authentication, data origin authentication, email source authentication, spam, phishing, security by integrity, digital signatures, undetected key compromise

1. INTRODUCTION AND OVERVIEW

One of the best-known security paradigms is to use authentication as the basis for access control decisions [4]. For example, to log in to a computer account, a user identifies an account by a *userid*, and enters a *password* as evidence of user authenticity; successfully authenticated users are granted access to various account resources as specified by a profile associated with that userid. More fine-grained access control commonly involves the use of access control matrices which specify permissions indexed by authenticated subject and object (resource), or equivalently, per-object access control lists specifying privileges associated with authenticated users. Turning this around, we propose an approach which uses access control (more specifically, integrity) as the basis for message authentication. Thus our proposal might be referred to as *security by integrity.* We propose a simple, practical means by which data origin assurances for message authentication are based on corroboration. The simplest example of this is by cross-checking with information posted at a specified location (e.g., web page) by a known or implied source. The security relies on the integrity of this corroborating information, and thus on access control in the hosting (or publishing) of this information. We assume that an adversary cannot modify data hosted at that site.

Our proposal is at heart non-cryptographic – we do not explicitly require cryptographic keys for either the corroboration step, or for the protection of corroborating information. For example, the latter can be publicly posted (although its authenticity and integrity is critical – similar to the situation for public keys). Because we do not require cryptographic keys explicitly, and require as a base assumption the integrity of independent corroborating data – which we hope is a reasonable practical assumption (as discussed later) – our work does not fit into mainstream cryptographic models.

Applications of our paradigm include any type of message sent, for which verification of origin and integrity are important (i.e., *source authentication*), spanning from news articles and press releases to instant messaging (IM). One application of particular note is email, including the problems of spam and phishing attacks, which often (but not always) involve some form of spoofed email source identity. While our proposal cannot eliminate spam or phishing – because it does not preclude attackers from using, e.g., their own machines and short-lifespan free email accounts (cf. [3]) – it can nonetheless help reduce email with spoofed source addressing. Moreover, clear view of the origin of spam and phishing email sent, i.e., that with unspoofed source addressing, may allow counteractive measures (e.g., source-based filtering, or notifying/confronting the legitimate owner of the machine) if the use of this source persists for significant durations.

Our proposal also has application to digital signatures, as determining the true (physical) origin of a digital signature provides important evidence for signature verification – which we believe has been largely overlooked to date. Our proposal helps determine precisely this, as opposed to conventional digital signature verification, which generally aims to verify that a signature in question almost certainly was created using a particular signature private key – whether or not that private key has been (possibly unknowingly) extracted, duplicated, or deduced by others.

One possibility for selectively deploying our proposal may be for mail servers to provide corroboration for a subset of mail messages (e.g., those of high value, according to some metric). One potential economic model may be for mail servers to charge a nominal fee (e.g., fractional pennies) for publishing information allowing corroboration of message origin.

The sequel is organized as follows. §2 presents a generic overview of our main proposal for verifying message (data origin) authenticity, and its applicability to two problems of high interest: email with spoofed source addressing (which is characteristic of a large proportion of spam and phishing email); and digital signatures (specifically with respect to the problem of compromised signature private keys). §3 provides further discussion, including regarding threat model. §4 reviews related work. §5 gives concluding remarks.

2. SECURITY BY INTEGRITY

In the subsections below, we present a high level overview of the new paradigm, and discuss two specific applications.

2.1 Generic version of message authentication proposal

Our proposal for message authentication (or equivalently, data origin authentication) is extremely simple. The main idea is as follows. Let A be the *originator* of a data message, which can be any digital string including, e.g., an email message or a digital signature itself. A causes the data message to be transmitted, or otherwise made available, to one or more message recipient(s) B. Also required to be made available in some form (as explained below) is an assertion of the purported originator of the message (i.e., an assertion that A originated it). The functionality of message authentication we wish to provide is that B be able to verify (with a reasonably high, but unspecified here, degree of assurance) the asserted source of the message.

One way for B to do this verification is to compute, from the received copy of the message, a fingerprint or message digest (e.g., using a cryptographic hash function), and verify it through an independent trusted channel (e.g., by phone to a recognized voice using A's known telephone number). This is a known technique for authenticating public keys, e.g., by reading out the digest in hex, over the phone. Our proposal involves in part a variation of this, using a more convenient automated Internet-based corroboration method – for example, posting the fingerprint at a "trusted" (as clarified by the requirements below), publicly available verification location for B to cross-check.

We list five requirements for our proposed method.

1. B must know the authentic location (e.g., URL) of the verification information. The integrity of this location information is critical.
2. A (and not others) must have the ability to cause the corroborating information to be published.
3. This information must be published in essentially real time (i.e., in time for B to verify).
4. This publishing should be convenient for A to arrange.
5. An attacker must not be able to publish or modify such published corroborating information, or alter responses to queries on it.[1]

Regarding this latter point, one threat considered in our model is an attacker who has compromised the computing device from which A originates messages. Ideally, the published corroboration information is part of a system whose access control mechanism is independent from this device; however, independence from user A herself is not possible, given that A should control publication to this location (see further discussion in §3).

A simple instantiation of this proposal involves every originator A having their own web site (web presence) W_A, and posting on W_A a fingerprint of (selected) messages they have originated. The site W_A associated with each potential originator A would have to be known a priori (or itself communicated in a secure manner, i.e., with guaranteed integrity and authenticity – much as for public keys in public key systems). While this is a non-trivial challenge, and one which we do not wish to under-estimate, we believe that this type of information may be established over time by out-of-band means, or e.g., by use of web sites uniquely associated with email addresses (the latter of course being necessarily unique); these same challenges arise in identity-based cryptosystems [29, 6].

2.2 Application no.1: detecting spoofed source addressing in email

As an application of the proposal of §2.1, we consider the problem of email with spoofed source identities. As our present objective is to propose an architectural design, we do not dwell on the very important (and surprisingly difficult) [20, 33, 10, 1] implementation detail of *which* source identities are best authenticated, such as the RFC 2821 [19] MAIL FROM domain (envelope sender/bounce address) or HELO identity (SMTP sending host); or the RFC 2822 [27] message header originator fields (which are unused by SMTP once the email begins transit) 'From:', 'Sender:', etc.

[1] Thus an authenticated channel is needed for publishing, and a channel with integrity is needed for retrieving, such information. This present architectural paper does not constrain how to implement such channels.

A significant subset of current Internet mail problems arises due to attackers having gained control of innocent users' machines (e.g., through exploitation of ubiquitous flaws in commodity software), and using such machines for sending unwanted email. Such email includes that to market generally unwanted products (*spam)*, and that directing users to click on links to spoofed web sites in an attempt (*phishing*) to extract users' personal information (e.g., userid-password information, credit card numbers, banking information, etc.) in an online version of social engineering.

Several current proposals to address spam and phishing involve the use of some form of public-key infrastructure (PKI). For context, later in this section we briefly overview two of these in actual deployment by many Internet domains:[2] SPF [20, 33] and DomainKeys/DKIM [10, 1].[3] Our proposal of §2.1 may be viewed as a simpler alternative to SPF and DomainKeys, which avoids some of their drawbacks, including: their breaking of certain mail forwarding and list server functionality; their additional burdening of the Internet's core DNS infrastructure; and the computational overhead inherent in digital signatures as used by DomainKeys. Under Case 2 (immediately below), our proposal also allows finer granularity of authentication than either SPF or DomainKeys, though the latter could theoretically compete. On the other hand, SPF and DomainKeys may have other advantages over our proposal – further research is required, as these proposals evolve.

Two possibile implementations of our generic proposal for this specific application are:

- Case 1: in each enterprise organization or ISP, the outgoing mail server maintains a per-email-address list of the hashes of all (or selected) outbound mail originated.
- Case 2: each user mail client maintains, for each email address they own, a list of the hashes of all (or selected) outbound mail originated.

These hash lists should be made available at publicly accessible web addresses, per email address, and e.g., searchable by message hash; hashes might be kept for a suitable window of time (e.g., 7 days, 90 days, or longer if necessary). It is unclear if storing these hashes for longer periods is necessary, if we view the primary purpose of our source authentication as allowing a simple verification (e.g., in a one-time check) that the asserted email source address information is correct, rather than to support non-repudiation.

Origination vs. (Re)transmission of Content. Our assertion of advantage over existing proposals which break certain mail forwarding and list server functionality requires further comment. To be fair, in its simplest form, our proposal addresses the problem of who (originally) authored the content of a message, avoiding implementation-dependent issues that arise in determining the authority of intermediate agents in a delivery network to forward, retransmit or relay such content. An advantage may in fact result from the choice to solve a simpler (and perhaps, more appropriate) problem. Indeed, one criticism of SPF and DomainKeys is the lack of clarity regarding (a) what problem they are actually designed to address; and (b) precisely what the threat model is. We attempt to clarify this somewhat for our proposal: for (a) see above and Q2 in §5; for (b) see §3.

Sender Policy Framework – simplified overview. SPF (Sender Policy Framework) [20, 33], evolved from predecessors DMP and RMX, has been supported by AOL among others; Microsoft has a Sender ID extension of it.[4] When email is sent from an originator A to a recipient B using standard SMTP, the TCP/IP connection established between mail transfer agents MTA_2 of the receiving domain and MTA_1 of the originating domain provides MTA_2 with the IP address IP_1 of MTA_1. SPF involves MTA_2 looking up in DNS a special SPF record associated with the domain of the email's asserted MAIL FROM and/or HELO identity [19], specifying all authorized originating MTA's of that domain. If IP_1 is not among these, then the email is flagged as questionable (and actions are taken as per the policy of the receiving system). One drawback of SPF is that it breaks certain types of inter-system mail forwarding.

SPF answers the question: *According to DNS-based SPF records, is the sending host (in the current SMTP session) authorized to send mail on behalf of the domain implied by the asserted MAIL FROM and/or SMTP HELO identity?* In contrast, our proposal answers the question: *Does the asserted originating mail server (Case 1), or asserted originating mail client (Case 2), confirm having originated this specific mail message?* We believe that these answers support the view that our proposal provides more precise information than SPF, while requiring neither that each domain create new DNS-based SPF records, nor the additional reliance on the DNS infrastructure for SPF queries.

DomainKeys – simplified overview. DomainKeys [10, 1] originated from Yahoo and is backed by Google, among others. For a given message M, DomainKeys involves the sending domain's SMTP server adding a (SHA-1 based RSA) digital signature over M into the SMTP header. Roughly speaking, the signature is over the mail contents, plus selected header fields not expected to be changed by transport agents. The receiving SMTP server retrieves from DNS the public key corresponding to the email's implied originating domain (or optionally, more specifically an originating entity therein), and uses it to verify the digital signature. Validity provides reasonable evidence that M did indeed originate from a person or system authorized to send mail from that domain. Mail servers may choose to automatically drop mail with invalid signatures. Cryptographic overhead has been voiced as one potential drawback of DomainKeys. Another is the breaking of some mail list servers (for example, if a 'Subject:' message header line is modified, or text is added to the end of the body). Another possible objection, by some, is a purported loss of the ability to repudiate casual

[2] It remains unclear if any, one, or both of these (complementary) protocols will be widely adopted.

[3] DomainKeys was the basis from which DKIM was derived; the former will become historical as the latter progresses. Hereafter, unless explicitly clarified, *DomainKeys* implies both generically, despite their evolving differences.

[4] A March 2, 2005 Microsoft release claimed over 750,000 (vs. 70 million existing) domains had published SPF records.

messages signed on one's behalf by one's local mail server (e.g., if the potential to repudiate is viewed as a privilege of private social conversation, including email).

2.3 Application no.2: compromised digital signature keys

As a second application of the proposal of §2.1, we consider the problem of stolen or otherwise compromised digital signature private keys. To our knowledge, this problem has received little formal attention in the research literature (see §4.1; our related work [16] proposed requiring a "second level of authentication" before accepting digital signatures, "based on information shared with a trusted authority").

To address this problem, consider a variation of the implementation of §2.2 whereby, rather than posting hashes of sent mail messages, what is posted are the actual bitstrings corresponding to all (or selected) originated outgoing (from legitimate client machines) digital signatures. We might also consider posting, as an alternative to a digital signature bitstring, any other function of the signed message which uniquely identifies the message (e.g., a cryptographic hash).

We believe this application of our proposal has the potential to simplify some important aspects of public-key infrastructures (PKI's) for digital signatures, specifically for PKI's intended to adequately address the difficult related issues of key revocation and potentially undetected signature private key compromise.[5] However, an important issue, with respect to signatures intended to provide the property of non-repudiation, is that a user (or organization) must be prevented from repudiating a signature by simply deleting the corresponding posted corroborating information from a site they control. Thus, in this application it is important to ensure that the corroborating site be an independent party[6] in the sense that any potential relying party would trust it not to delete corroborating information if so requested by an originator (who seeks to repudiate). One option is to use an unerasable log file.

We believe that an important question that should be in the mind of a relying party in digital signature verification is: *was it the asserted party (or someone else) who actually used the private key to sign this message,* rather than: *who is officially associated with the private key used?* Regarding the question of private keys extracted from client software, and the related issue of independence of the corroborating web site, see additional discussion in §3.

The application of our proposal to address this problem of potentially compromised digital signature private keys requires further exploration.

3. THREAT MODEL AND FURTHER DISCUSSION

Among others, two problems our proposal addresses are:

Problem A: spoofing of email origination addresses from a distinct machine; and

Problem B: extracting a signature key and using it remotely to forge signatures.

A common characteristic in both is a type of "spoofing" where essentially an attacker is making an assertion of being a message originator, leaving the actual legitimate (spoofed) party to find some way to detect and/or disprove the assertion. We now consider attack models.

MODEL 1. The simplest attack model assumes that the attacker does *not* control, or have access to, the spoofed machine (case 1-A) or that on which legitimate signatures are normally created (case 1-B).[7] Our proposal addresses both these cases, and for this model, it is not required that access control information for the corroborating site be stored independently from the machine normally originating messages. A convenient implementation might involve a user's machine automatically triggering the update of information on the corroborating site.

A more challenging scenario under Model 1, is case 1-Ax: an attacker who sends mail (e.g., spam or phishing) *without* spoofing source addresses, instead, say, using his own personal machines and accounts. Our proposal cannot address this, because for example such an attacker can control his own "legitimate" corroboration sites. However, proposals such as SPF and DomainKeys likewise cannot address this scenario.

MODEL 2. Another model, more favourable to the attacker, involves compromised machines as follows. Case 2-A: an attacker uses compromised machines, and originates email messages (e.g., spam or phishing) using source email addressing of the compromised machine; note that technically, this is *without* spoofed addressing. Case 2-B: an attacker has one-time access to a victim's machine, allowing theft of a signature private key (2-B1);[8] or continued access to a victim machine (2-B2). Under Model 2, for our proposal, access control information for the corroborating site must indeed be controlled independently from the (compromised) machine normally originating messages. Under such independence, our proposal addresses the threat.[9] Otherwise, if access to the corroborating site is controlled through e.g., a password stored on a user's machine, then the corroborating site does not offer truly independent corroroboration.

In case 2-A, if such access control information is not independently controlled, our proposal may still give defenders a chance to detect and respond, as the traceability of email may allow identification of compromised nodes. A solid start towards reducing spam and phishing should result from being better able to answer the question: *Which machine actually originated this message?*

[5] We note that to date, the vast majority of PKI's deployed in practice have been used for encryption and authentication services, vs. digital signatures with non-repudiation.

[6] The functionality required of this party should be compared in greater detail to requirements of a digital notary.

[7] For case 1-B, we nonetheless assume that the attacker has a victim's signature private key, e.g., extracted through a *side-channel* attack such as timing analysis [7] not involving compromise of the victim's machine itself.

[8] Digital theft is generally difficult to detect, since copying bitstrings leaves no traces.

[9] Note that SPF and DomainKeys do not address case 2-A.

Model 2 seems quite plausible in today's Internet environment of ubiquitous malware being able to exploit countless vulnerabilities in commodity software. But for access control information maintained independently of a victim machine, under our proposal, the attacker's task is considerably more difficult: if such access control information is not also independently stolen, then for each case 2-B1 or 2-B2 signature successfully forged, an attacker would require the ability to either compromise the integrity of the corroborating site, or have control over the data posted to it (either permanently, or through regular re-access).

The above discussion raises the following practical question to be pursued, to better understand what implementation choices to make for our proposal: *What is the need for (i.e., what threat model is of main concern), by what means can we achieve, and to what degree can we assure, integrity in the sense of access control on a corroborating site?*[10]

4. RELATED WORK

In subsections below we discuss related work under three categories: security through collaboration and independent corroboration, keyless cryptography, and non-cryptographic keyless authentication mechanisms. While not all of this work relates directly, we believe it provides good context for positioning our proposal as a new security paradigm. Additional related work is discussed in subsections of §2.

4.1 Collaboration and independent corroboration

The ideas of separation of duty (e.g., [31]), split control, secret sharing and threshold schemes are well-known in the literature (e.g., Desmedt [11]; see Menezes et al. [21, pp.538-539] for further references). The security benefits of using cross-checking or corroboration to remove single points of failure are also well-known. For example, C. Kahn [17] lists numerous "independent corroboration" approaches for achieving system resilience in the presence of unreliable (possibly compromised) components, including: voting among independent agents, interactive consistency checks, state machine replication (using multiple identical machines), redundant webs of trust or trust meshes [25] (see also Reiter [26] re: corroborating credentials, and Fritz [13] re: peer-to-peer distributed authentication of public keys), direct verification of assertions by relying parties, and cooperation among peers to recognize and report misbehaving neighbours. Another long-standing proposal, the *Merkle channel* [30, p.387], involves distributing public keys over a sufficiently large number of independent channels (print media, television, etc.) to make it infeasible for an attacker to tamper them all.[11] Perlman's robust flooding [23] may be viewed as a somewhat related idea, itself motivated by the Byzantine Generals Problem, and part of the vast literature on Byzantine protocols. The idea of security by integrity also appears in work by Haber et al. [15, §2.4], where the trustworthiness of certificates relies on the integrity of a repository.

[10] Cohen [8] has noted that ultimately, all forms of security are based on either physical access, or specific knowledge of details (e.g., including keys) of access procedures.

[11] Of all related work discussed, this is perhaps the most closely related in spirit to our own, even though this earlier work does not address the same problems, and we do not use public keys.

Related to our digital signature application, Blakley et al. [5] discuss the idea of both a signer and a registrar recording all generated digital signatures, with the signer protecting associated keying matter "in a manner completely independent of the way he hides his [secret signing key]". Just et al. [16] mention the idea of "recording user signatures in an integrity-protected database" or third-party *trust register*.

4.2 Keyless cryptography – puzzles and information hiding

In the past, a number of secure communications systems have been positioned as *keyless cryptography*, i.e., providing some form of security (e.g., confidentiality in the sense of encryption) without the use of secret keys.[12] For example, Kahn [18] notes a set of cryptographic schemes circa 1772 by F.J. Buck, which have recently been characterized as having the normal requirement of secret keys replaced by the ability to solve specific classes of puzzles ("it is a game of hiding messages inside algebraic puzzles" [14]). The security of these schemes rests in the lack of knowledge of the system details by an adversary – contradicting Kerckhoffs' fundamental assumption of cryptography. *Steganography* (e.g., see [24]) involves algorithms for hiding information wherein it is a goal to obscure even knowledge of the presence of a hidden message, typically transmitted under the *cover* of other data. Many modern steganographic schemes now also involve the use of secret keys.

In our own earlier cryptographic research (e.g., [21]), we have come across few schemes which provide security without keys. While several cryptographic schemes have been implicitly assumed to be, or explicitly referred to as *keyless*, these typically still involve some form of key. Simple Diffie-Hellman key exchange begins with two parties sharing no private or public keys, but the exponentials exchanged are (unauthenticated) public keys, with corresponding private keys known only to the respective parties. The so-called keyless cryptography scheme of Alpern and Schneider [2] does in fact involve secret keys, namely the $2n$-bit strings which each party must not reveal publicly. Shamir's "no-key protocol" [21, p.500] requires that each party select a symmetric (secret) key not disclosed to the other. Merkle's puzzle system [22] may be viewed as involving a fixed number (n) of keys, which the two legitimate parties Bob and Alice each separately must exhaustively search over, while imposing a work factor of n^2 on an adversary.

4.3 Non-cryptographic (keyless) authentication mechanisms

Many known techniques for providing information security do not involve cryptographic keys. One obvious example is the storage of sensitive documents (physical or digital) in physically controlled spaces such as filing cabinets in locked offices. Keyless security solutions which provide some form of (public) authentication are of greater relevance to our present work. In this section we discuss a few examples, in particular those employing an independent trusted channel or a known address.

[12] The presence of a secret key is a typical, albeit imperfect, distinguisher of cryptographic schemes.

As a first example, the provision of data integrity by combining an unkeyed hash function with some form of authentic channel is an alternative to using message authentication codes (e.g., see [21, p.364]; cf. the Merkle channel above). As a second example, web sites requiring passwords for access will now commonly send to a user's email address on record, an email message containing their (forgotten or intial) password; the assumption is that this typically unencrypted email channel is inaccessible to an attacker. Similarly, credit cards, credit card PIN numbers, and account passwords may be distributed by the postal mail system, or internal corporate mail systems, under the assumption that such mechanisms offer some form of trusted delivery (e.g., physical or procedural). Activation of some credit cards requires calling a toll-free number from the legitimate owner's home telephone number on record; this relies upon the "integrity of telephone system", assuming that there is greater risk for an attacker to call from the victim's home residence, and that it is relatively difficult to spoof caller-ID information on the wire if calling from elsewhere. The assumed integrity of the phone system is similarly relied upon by authentication schemes which involve call-back to a home or office phone number on record (or to a cell phone, including as a text or SMS message). For cell phones or other portable devices, an additional assumption is that it is difficult for an attacker to steal a victim's cell phone. Call-backs may be used as a second-factor of authentication, and commercial such services are available (e.g., http://www.authentify.com).[13] In another bricks-and-mortar example, card-not-present credit card transactions which involve the purchase of physical goods to be delivered, typically require that the delivery address be that on record as the credit card owner's address.

5. CONCLUDING REMARKS

A fundamental question arising from our work is as follows.

Q1: *What message authentication guarantees can we get without direct reliance on secret keys?*

The types of guarantees we have in mind are source (data origin) verification, to an assurance level suitable to address a meaningful portion of commonplace practical threats. A more practical question arises from our proposal, and our §2.2 remark re: avoiding delivery-related issues.

Q2: *Can the proposal be adopted to handle issues related to message forwarding, retransmission, relay, and/or replay?*

We suspect that techniques which work for DomainKeys may be candidates to attempt to adapt for our proposal.

Cryptography has been of great benefit in securing the Internet, but drawbacks of cryptographic solutions often include complexity, interoperability, and application integration (e.g., as experienced by those attempting to deploy PKI). In many applications, cryptographic solutions are also computationally expensive – not e.g., for 2GHz desktop machines for infrequent user-triggered digital signatures, but rather e.g., at servers and for applications such as frequent automated infrastructure or maintenance messages, individual records and fields in databases, and routing update messages. Moreover, crytographic solutions (and indeed, most security solutions) are extremely difficult to retrofit. We view our proposal as a "featherweight" solution to data origin authentication, which scores favourably on all of the above issues, without requiring direct modifications to existing infrastructure (such as the DNS record system [10]).

We believe the paradigm of independent corroboration is under-utilized in today's practice of Internet security, although this may change with increasing use of P2P solutions. Our proposal is a varation of previous forms of independent corroboration, whose time we believe is ripe; it would not have been practical on a broad scale even 10 years ago, without the widespread connectivity enjoyed by today's Internet, the ability of almost anyone to make information publicly available, and the low cost and ubiquity of online verification.

Our experience is that highly complex systems (including many which on paper are "provably secure") do not make their way into the real world often, and when they do, they are too frequently easily side-stepped by attackers for a number of reasons (none of which may directly contradict the security claims of the system designers). Thus, for practical deployment, we favour the alternative of simple, lightweight approaches – because they are far more likely to be deployed, and even if not offering national-security level strength, they are of tremendous incremental value over the status quo. We believe our proposal offers such an alternative for data origin authentication in distributed open systems. Our proposal is also a natural opposite of *security by obscurity*, by promoting *security through publicity*.

Our proposal as detailed herein is high-level, preliminary, and may have drawbacks unrecognized as yet. It will benefit from detailed formulation and critical review. To this end, our motivation is to stimulate broad discussion, in order to advance, and adjust the approach as necessary, to enable adoption in practice.

Acknowledgements. Sincere thanks to everyone at NSPW in Lake Arrowhead for lively and insightful discussion of this work, to Bob Blakley and Carrie Gates for their detailed notes of that discussion, and to members of Carleton's Digital Security Group for discussion and feedback. I thankfully acknowledge Canada's Natural Sciences and Engineering Research Council (NSERC) for funding under a Canada Research Chair in Network and Software Security, and an NSERC Discovery Grant.

6. REFERENCES

[1] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, M. Thomas, "DomainKeys Identified Mail (DKIM)", Internet Draft (work in progress), draft-allman-dkim-base-00 (July 9, 2005).

[2] B. Alpern, F. Schneider, "Key Exchange Using 'Keyless Cryptography' ", *Information Processing Letters* 16 (1983), 79–81.

[3] S. Bellovin, "Spamming, Phishing, Authentication and Privacy", Inside Risks, *C.ACM* 47(12), Dec.2004.

[13] Phone call-backs may also be used to convey one-time passwords (short-term secrets), but recall our focus here is keyless solutions.

[4] M. Bishop, *Computer Security: Art and Science*, 2002, Addison-Wesley.

[5] B. Blakley, G.R. Blakley, "All Sail, No Anchor II: Acceptable High-End PKI", *International Journal of Information Security* (2004) 2:66-77.

[6] D. Boneh, M. Franklin, "Identity-Based Encryption from the Weil Pairing", *SIAM Journal on Computing* 32(3), 586–615, 2003.

[7] D. Brumley, D. Boneh, "Remote timing attacks are practical", USENIX Security 2003.

[8] F. Cohen, "Operating System Protection Through Program Evolution", *Computers and Security* 12(6), 1 Oct. 1993, pp. 565–584.

[9] C. Collberg, C. Thomborson, D. Low, "A Taxonomy of Obfuscating Transformations", Tech. Rep. 148, Dept. Computer Science, Univ. of Auckland (July 1997).

[10] M. Delany, "Domain-based Email Authentication Using Public-Keys Advertised in the DNS (DomainKeys)", Internet Draft (work in progress), draft-delany-domainkeys-base-03.txt (29 Sept. 2005). See also: http://en.wikipedia.org/wiki/Domain-keys.

[11] Y. Desmedt, "Threshold Cryptography", *Euro. Trans. Telecom.* 5(4) 449–457, July-August 1994.

[12] S. Forrest, A. Somayaji, D. H. Ackley, "Building Diverse Computer Systems", pp. 67–72, Proc. 6th Workshop on Hot Topics in Operating Systems, IEEE Computer Society Press, 1997.

[13] A. Fritz, J.-F. Paris, "Maille Authentication: A Novel Protocol for Distributed Authentication", Proc. 19th IFIP Information Security Conference (SEC 2004), Toulouse, France, Aug. 2004.

[14] J. von zur Gathen, "Friederich Johann Buck: Arithmetic Puzzles in Cryptography", *Cryptologia*, Oct. 2004.

[15] S. Haber, W.S. Stornetta, "Secure Names for Bit-Strings", 4th ACM Conference on Computer and Communications Security (CCS'97), 1997.

[16] M. Just, P. van Oorschot "Addressing the Problem of Undetected Signature Key Compromise", *Proceedings of the Network and Distributed System Security Symposium*, NDSS 1999, San Diego, California, The Internet Society 1999.

[17] C. Kahn, "Tolerating Penetrations and Insider Attacks by Requiring Independent Corroboration", 1998 New Security Paradigms Workshop (NSPW'98). (See also same author: "Using Independent Corroboration to Achieve Compromise Tolerance", 1998 Information Survivability Workhop (ISW) 1998.)

[18] D. Kahn, *The Codebreakers*, MacMillan, 1967.

[19] J. Klensin, "RFC 2821 - Simple Mail Transfer Protocol", IETF (Standards Track) Request for Comments, April 2001.

[20] M. Lentczner, M. Wong, "Sender Policy Framework: Authorizing Use of Domains in MAIL FROM", Internet Draft (work in progress), draft-lentczner-spf-00 (Oct. 2004). See also: http://en.wikipedia.org/wiki/Sender-Policy-Framework.

[21] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996. Online: http://www.cacr.math.uwaterloo.ca/hac/

[22] R. Merkle, "Secure Communications over Insecure Channels", *C. ACM* 21 (1978), 294–299.

[23] R. Perlman, *Network Layer Protocols with Byzantine Robustness*, MIT LCS TR-429, Oct. 1988.

[24] F. Petitcolas, R.J. Anderson, M.G. Kuhn, "Information Hiding – A Survey", *Proc. of the IEEE* (Special Issue on Protection of Multimedia Content), vol.87 no.7 (July 1999), pp.1062–1078.

[25] M. Reiter, S. Stubblebine, "Toward acceptable metrics of authentication", 1997 IEEE Symposium on Security and Privacy.

[26] M.K. Reiter, S.G. Stubblebine, "Path independence for authentication in large-scale systems", 4th ACM Conference on Computer and Communications Security (CCS'97), 1997.

[27] P. Resnick, "RFC 2822 - Internet Message Format", IETF (Standards Track) Request for Comments 2822, April 2001.

[28] T. Sander, C.F. Tschudin, "Protecting Mobile Agents Against Malicious Hosts", pp. 44–60, *Mobile Agents and Security*, Springer LNCS 1419 (1998).

[29] A. Shamir, "Identity-based cryptosystems and signature schemes", Crypto'84 (LNCS 196), 47–53, 1985.

[30] G. Simmons, "A survey of information authentication", G. Simmon (ed.), *Contemporary Cryptography: The Science of Information Integrity*, 379–419, IEEE Press, 1992.

[31] R. Simon, M. Zurko, "Separation of Duty in Role-Based Environments", Proc. 1996 Computer Security Foundations Workshop.

[32] Trusted Computing Group, http://www.trustedcomputinggroup.org/home.

[33] M. Wong, W. Schlitt, "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-MAIL, version 1", Internet Draft (work in progress), draft-schlitt-spf-classic-02 (June 6, 2005).