Message Authentication by Integrity with Public Corroboration*

P.C. van Oorschot

School of Computer Science, Carleton University, Canada
paulv@scs.carleton.ca

Abstract. One of the best-known security paradigms is to use authentication as the basis for access control decisions. We turn this paradigm 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 posted 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 in the hosting (or posting) 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*. Example types of message authentication we discuss include email source origination, and data origin authentication for digital signatures. Our work thus has application to problems including spam and phishing (email with spoofed source addresses), and theft, extraction, or other determination of digital signature private keys.

1 Introduction and Overview

One of the best-known security paradigms is to use authentication as the basis for access control decisions [3]. For example, to login 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 posting) of this information. We assume that an adversary cannot modify data hosted at that site.

* Version: 19 April 2005.

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. Indeed, in the simplest example, the latter is 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*), e.g. news articles, press releases, and instant messaging (IM). One application of particular note is email, including the problems of spam and phishing attacks, which generally involve spoofed 'From' addresses in email headers. While our proposal can not eliminate spam or phishing – because it does not preclude attackers from continuing to use compromised machines with true 'From' addresses, and also because "spammers can authenticate themselves too" [2] – it can nonetheless help eliminate email messages with spoofed 'From' addresses. Moreover, the clear origin of spam and phishing email sent with non-spoofed 'From' addresses allows 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.

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 from addresses (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. §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. The data message 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 other parties B, the message recipient(s). 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 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 four important requirements here.

1. B must know the location (e.g. URL) of the verification information. The integrity of this location info is critical (cf. the integrity of a public key in a public key system).
2. A must have the ability to cause the corroborating information to be posted or made available, and this must be possible in essentially real time (i.e. in time for B to verify).
3. The posting of corroborating information should be easy for A to arrange (ideally, automated – but not so much that the next point is violated).
4. An attacker must not be able to post or modify such corroborating information.

Regarding this latter point, it is important to consider the possibility that the attacker has compromised the computing device itself from which A originates messages. In general, the public verification location is ideally an independent system with separate access control – however, complete independence from the user A (as opposed to from A's originating computer) is not possible, given the requirement that A must be able to cause updates to this location. For further discussion, see Section 3.

A simple instantiation of this proposal involves every originator A having their own web site (web presence) W_A, and posting a fingerprint of messages they originated on W_A. 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) – again much as is the case for public keys in certificate-based systems.[1] While this is a

[1] Note the resemblance to identity-based cryptosystems [24, 4].

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), much as these same challenges are being addressed in relation to identity-based cryptosystems [24, 4].

2.2 Application No.1: Detecting spoofed email 'From' fields

As an application of the proposal of Section 2.1, we consider the problem of email with spoofed 'From' fields. A non-trivial fraction of problems with current Internet email arises from attackers who have gained control of innocent users' machines (e.g. through exploitation of ubiquitous flaws in commodity software), and use 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 [16] and DomainKeys [8].

Our proposal of Section 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 infrastructure, DNS; 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. However, SPF and DomainKeys may also have advantages over our proposal – further analysis is required here.

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 outbound email originated.

Case 2: each user mail client maintains, for each email address they own, a list of the hashes of all outbound email 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 day, 90 days, or longer if necessary). However, 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 'From' field of email is correct, rather than to support non-repudiation.

[2] It remains unclear if either will eventually become dominant. One view is that SPF and DomainKeys will be deployed alongside each other; by design, they are compatible.

Sender Policy Framework (SPF) – simplified overview. Sender Policy Framework [16] evolved from predecessors DMP and RMX. AOL supports SPF, as does Microsoft through their own Sender ID extension of it.[3] 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 purported 'From' address, 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 inter-system SMTP forwarding.

SPF answers the question: *According to the new DNS-based SPF record, is the originating MTA authorized to send mail on behalf of the domain corresponding to the purported 'From' address?* In contrast, use of our proposal answers the question: *Did the originating MTA (in Case 1), or the originating email client (in Case 2), actually send this specific email 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 [8] originated from Yahoo and is now strongly backed by Google, among others. For a given email message M, DomainKeys involves the sending domain's SMTP server adding a (SHA-1 based RSA) digital signature over M into the SMTP header. The receiving SMTP server retrieves from DNS the public key corresponding to the email's purported originating domain, and uses it to verify the digital signature (providing strong evidence that M did indeed originate from the claimed domain). Mail servers may choose to automatically drop mail with invalid signatures. One drawback of DomainKeys is that it breaks some forwarding mechanisms (e.g. some list server modifications). Among other objections, some have cited the loss of the ability to repudiate messages signed on one's behalf by one's local mail server; the potential to repudiate is viewed by some as an advantage or privilege of private social conversation (including email).

2.3 Application No.2: Compromised Digital Signature Private Keys

As a second application of the proposal of Section 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, aside from our own previous work [13], which 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 Section 2.2 whereby, rather than posting hashes of sent mail messages, what is posted are the actual bitstrings corresponding to all originated (outgoing) digital signatures created. We might

[3] A March 2, 2005 Microsoft press release stated that over 750,000 domains had published SPF records.

also considering posting (as an alternative to a digital signature bitstring) a 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.[4] 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 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). Recall that an important question in the mind of a relying party carrying out digital signature verification is: *Did the asserted signing party really sign this message (and when)?*

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 Section 3.

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

3 Further Discussion

Among possibly others, two specific problems our proposal addresses are:

Problem A: spoofing of email 'From' 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 disprove, the assertion. Our attack model assumes that the attacker does *not* typically control the spoofed machine (Problem A) or that on which legitimate signatures are created (Problem B). We note that proposals such as SPF [16] and DomainKeys [8] likewise do not prevent compromised machines from continuing to be used e.g. for directly originating spam. However, we now briefly consider whether our proposal has any possible value in an attack model more favourable to the attacker.

Even in the situation of an attacker having continued use of (or recurring access to) a victim machine (hereafter, *Situation X*), our proposal may give defenders a better chance to detect attacks, if better traceability and accountability for malicious messages helps identify compromised nodes. A solid start towards reducing email spoofing and phishing, and digital signature forgery, should result from being better able to answer the question: *Did this machine truly originate this message?* Regarding the likelihood of Situation X, we

[4] Note that to date, the vast majority of PKI's deployed in practice have been used for encryption and authentication services, rather than digital signatures for non-repudiation.

expect it to be hard, but not impossible, for an attacker to maintain continous control and access to a computer, and to go undetected in modifying data posted for public verification.

Nonetheless, Situation X is considerably more difficult, especially for Problem B. Suppose a user's signature private key is extracted and duplicated by an attacker through a one-time access of the user's (i.e. victim's) machine. Such digital theft is generally difficult to detect, since copying bitstrings leaves no traces.[5] With our proposal in place, to successfully forge a signature, beyond possession of this private key, for each such signature, the attacker would require the ability to compromise the integrity of the corroborating site, or have control over the data posted to it (either permanently, or through regular re-access). Unless otherwise specified, a reasonable implementation of our proposal might involve the primary (legitimate) machine itself triggering the update of information on the corroborating site. Obviously, if access to the corroborating site is controlled through e.g. a password which is stored on this primary machine, then the corroborating site does not offer a truly independent corroroboration. Thus, we must ask: *What is the actual requirement, and to what degree can we guarantee, integrity in the sense of access control to a corroborating site?*[6] This question requires further pursuit to better understand our proposal in Case B under Situation X.

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 directly relates to our proposal, we believe it provides good context, particularly in light of the objectives of the New Security Paradigms Workshop. Additional related work is discussed in subsections of Section 2.

4.1 Collaboration and Independent Corroboration

The ideas of separation of duty (e.g. [26]), split control, secret sharing and threshold schemes are well-known in the literature (e.g. Desmedt [9]; see Menezes et al. [17, 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 [14] 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

[5] If a signature private key is instead extracted through a *side-channel* (e.g. timing analysis) attack [5] not involving compromise of the victim's machine itself, then our proposal stops forged signatures directly.

[6] Cohen [6] 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.

machines), redundant webs of trust or trust meshes [21] (see also Reiter [22] re: corroborating credentials, and Fritz [11] 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* [25, 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.[7] Perlman's robust flooding [19] may be viewed as a somewhat related idea, itself motivated by the Byzantine Generals Problem, and part of the vast literature on Byzantine protocols.

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.[8] For example, Kahn [15] 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" [12]). 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 [20]) 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. [17]), 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 [1] does in fact involve secret keys, namely the $2n$-bit strings which each party must not reveal publicly. Shamir's "no-key protocol" [17, p.500] requires that each party select a symmetric (secret) key not disclosed to the other. Merkle's puzzle system [18] 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.

[7] Of all related work discussed, we view this to be 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.

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

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.

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 [17, 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).[9] 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 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.

Cryptography has been of great benefit in securing the Internet, but drawbacks of cryptographic solutions often include complexity, interoperability, and application integration (e.g.

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

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 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 [8]).

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 such independent corroboration, whose time we believe is ripe; it would not have been practical on a broad scale even 10 years ago, prior to the widespread connectivity provided by today's Internet, the ability of almost anyone to make information publicly available, 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*.

Finally, we believe our proposal will be of particular interest at this time, due to its applicability to the currently high-profile problems of spam and phishing with spoofed 'From' addresses (cf. [8, 16]), among others.

We acknowledge that at present, the proposal as detailed herein is preliminary, incomplete, and requires significant additional formulation and critical review. It also likely has a number of drawbacks which have not been explicitly recognized, and it has not been implemented as yet even in a proof-of-concept. However, these issues motivate our submission of this work, in this (admittedly draft) form, to the New Security Paradigms Workshop for discussion.

References

1. B. Alpern, F. Schneider, "Key Exchange Using 'Keyless Cryptography' ", *Information Processing Letters* 16 (1983), 79–81.
2. S. Bellovin, "Spamming, Phishing, Authentication and Privacy", Inside Risks, *C.ACM* 47(12), Dec.2004.
3. M. Bishop, *Computer Security: Art and Science*, 2002, Addison-Wesley.
4. D. Boneh, M. Franklin, "Identity-Based Encryption from the Weil Pairing", *SIAM Journal on Computing* 32(3), 586–615, 2003.

5. D. Brumley, D. Boneh, "Remote timing attacks are practical", USENIX Security 2003.
6. F. Cohen, "Operating System Protection Through Program Evolution", *Computers and Security* 12(6), 1 Oct. 1993, pp. 565–584.
7. C. Collberg, C. Thomborson, D. Low, "A Taxonomy of Obfuscating Transformations", Technical Report 148, Dept. Computer Science, University of Auckland (July 1997).
8. M. Delany, "Domain-based Email Authentication Using Public-Keys Advertised in the DNS (DomainKeys)", Internet Draft (work in progress), draft-delany-domainkeys-base-02.txt (25 March 2005). See also: http://en.wikipedia.org/wiki/Domain-keys.
9. Y. Desmedt, "Threshold Cryptography", *European Transactions on Telecommunications* 5(4) 449–457, July-August 1994.
10. 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.
11. 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.
12. J. von zur Gathen, "Friederich Johann Buck: Arithmetic Puzzles in Cryptography", *Cryptologia*, Oct. 2004.
13. M. Just, P. van Oorschot "Addressing the Problem of Undetected Signature Key Compromise", Internet Society 1999 Symp. on Network and Distributed Systems Security (NDSS'99).
14. C. Kahn, "Tolerating Penetrations and Insider Attacks by Requiring Independent Corroboration", 1998 New Security Paradigms Workshop (NSPW'98). (See also: C. Kahn, "Using Independent Corroboration to Achieve Compromise Tolerance", 1998 Information Survivability Workhop (ISW) 1998.)
15. D. Kahn, *The Codebreakers*, MacMillan, 1967.
16. 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.
17. A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996. Online access: http://www.cacr.math.uwaterloo.ca/hac/
18. R. Merkle, "Secure Communications over Insecure Channels", *C. ACM* 21 (1978), 294–299.
19. R. Perlman, *Network Layer Protocols with Byzantine Robustness*, Ph.D. thesis, MIT LCS TR-429, October 1988.
20. 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.
21. M. Reiter, S. Stubblebine, "Toward acceptable metrics of authentication", 1997 IEEE Symp. Security and Privacy.
22. M.K. Reiter, S.G. Stubblebine, "Path independence for authentication in large-scale systems", Proc. 4th ACM Conference on Computer and Communications Security.
23. T. Sander, C.F. Tschudin, "Protecting Mobile Agents Against Malicious Hosts", pp. 44–60 in *Mobile Agents and Security*, G. Vigna (ed.), Springer LNCS 1419 (1998).
24. A. Shamir, "Identity-based cryptosystems and signature schemes", Crypto'84 (LNCS 196), 47–53, 1985.
25. G. Simmons, "A survey of information authentication", G. Simmon (ed.), *Contemporary Cryptography: The Science of Information Integrity*, 379–419, IEEE Press, 1992.
26. R. Simon, M. Zurko, "Separation of Duty in Role-Based Environments", Proc. 1996 Computer Security Foundations Workshop.
27. Trusted Computing Group, http://www.trustedcomputinggroup.org/home.