UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MOBILEIRON, INCORPORATED,
Petitioner

v.

BLACKBERRY, CORPORATION, *et al.*
Patent Owner

Case IPR2020-
U.S. Patent No. 7,372,961

DECLARATION OF SYLVIA HALL-ELLIS, PH.D. IN SUPPORT OF PETITIONER'S PETITION FOR *INTER PARTES* REVIEW

I, Sylvia D. Hall-Ellis, Ph.D., declare as follows:

I. INTRODUCTION

1. My name is Sylvia D. Hall-Ellis. I have been retained as an expert by MobileIron, Incorporated ("the Petitioner") who I am informed is the Petitioner seeking for the Patent Trial and Appeal Board to institute *inter partes* review (IPR) proceeding.

2. I have written this declaration at the request of the Petitioner to provide my expert opinion regarding the authenticity and public availability of publications, identified in Section V below. My declaration sets forth my opinions in detail and provides the basis for my opinions regarding the authenticity and public availability of these publications. If called to testify in the above-captioned matter, I will testify with regard to the opinions and bases set forth below.

3. I reserve the right to supplement or amend my opinions, and bases for them, in response to any additional evidence, testimony, discovery, argument, and/or other additional information that may be provided to me after the date of this declaration.

4. As of the preparation and signing of this declaration, libraries across the nation are closed pursuant to an order of the federal and state governments due to the COVID-19 virus. However, were the libraries open, I would expect to be able to obtain paper copies of the documents in this declaration. Additionally, it is

my typical practice to obtain a paper copy of each publication to further confirm my opinions that the documents were available prior to the alleged priority date of a patent under consideration. I reserve the right to supplement my declaration when the libraries reopen to provide such information.

5. I am being compensated for my time spent working on this matter at my normal consulting rate of $300 per hour, plus reimbursement for any additional reasonable expenses. My compensation is not in any way tied to the content of this report, the substance of my opinions, or the outcome of this proceeding. I have no other interests in this proceeding or with any of the parties.

6. All of the materials that I considered and relied upon are discussed explicitly in this declaration.

II. QUALIFICATIONS

7. I am currently an Adjunct Professor in the School of Information at San José State University in San José, California. I obtained a Master of Library Science from the University of North Texas in 1972 and a Ph.D. in Library Science from the University of Pittsburgh in 1985. Over the last fifty years, I have held various positions in the field of library and information resources. I was first employed as a librarian in 1966 and have been involved in the field of library sciences since, holding numerous positions.

8. I am a member of the American Library Association (ALA) and its

Association for Library Collections & Technical Services (ALCTS) Division, and I served on the Committee on Cataloging: Resource and Description (which wrote the new cataloging rules) and as the chair of the Committee for Education and Training of Catalogers and the Competencies and Education for a Career in Cataloging Interest Group. I also served as the Chair of the ALCTS Division's Task Force on Competencies and Education for a Career in Cataloging. Additionally, I have served as the Chair for the ALA Office of Diversity's Committee on Diversity, as a member of the REFORMA National Board of Directors, and as a member of the Editorial Board for the ALCTS premier cataloging journal, *Library Resources and Technical Services*. Currently I serve as a Co-Chair for the Library Research Round Table of the American Library Association.

9. I have also given over one hundred presentations in the field, including several on library cataloging systems and Machine-Readable Cataloging ("MARC") standards. My current research interests include library cataloging systems, metadata, and organization of electronic resources.

10. I have been deposed fourteen times: (1) *Symantec Corp. v. Finjan*, *Inc.,* Petition for *Inter Partes Review* of U.S. Patent No. 7,613,926, May 26, 2016, on behalf of Symantec Corp.; (2) *Symantec Corp. v. Finjan*, *Inc.,* 14-cv-299-HSG (N.D. Cal.), on behalf of Symantec Corp., September 14, 2017; (3) one deposition

for ten matters: *Intellectual Ventures I LLC v. AT&T Mobility LLC; AT&T Mobility II LLC, New Cingular Wireless Services, Inc., SBC Internet Services, Inc., Wayport, Inc., and Cricket Wireless LLC*, C.A. No. 12-193 (LPS); *Intellectual Ventures II LLC v. AT&T Mobility LLC; AT&T Mobility II LLC, New Cingular Wireless Services, Inc., SBC Internet Services, Inc., Wayport, Inc., and Cricket Wireless LLC*, C.A. No. 13-1631 (LPS); *Intellectual Ventures I LLC v. T-Mobile USA, Inc. and T-Mobile US, Inc.,* C.A. No. 13-1632 (LPS); *Intellectual Ventures II LLC v. T-Mobile USA, Inc. and T-Mobile US, Inc.,* C.A. No. 13-1633 (LPS); *Intellectual Ventures I LLC, v. Nextel Operations, Inc., Sprint Spectrum L.P., Boost Mobile, LLC and Virgin Mobile USA, L.P.,* C.A. No. 13-1634 (LPS); *Intellectual Ventures II LLC v. Nextel Operations, Inc., Sprint Spectrum L.P., Boost Mobile, LLC and Virgin Mobile USA, L.P.,* C.A. No. 13-1635 (LPS); *Intellectual Ventures I LLC, v. United States Cellular Corporation*, C.A. No. 13-1636 (LPS); *Intellectual Ventures I LLC v. United States Cellular Corporation*, C.A. No. 13-1637 (LPS); *Intellectual Ventures II LLC v. AT&T Mobility LLC*, *AT&T Mobility II LLC, New Cingular Wireless Services, Inc.,* C.A. No. 15-799 (LPS); *Intellectual Ventures I LLC v. T-Mobile USA, Inc. and T-Mobile US, Inc.,* C.A. No. 15-800 (LPS), on behalf of AT&T Mobility LLC; AT&T Mobility II LLC, Boost Mobile, LLC Cricket Wireless LLC, Nextel Operations, Inc., New Cingular Wireless Services, Inc., SBC Internet Services, Inc., Sprint Spectrum L.P., T-Mobile USA, Inc., T-

Mobile US, Inc., United States Cellular Corporation Virgin Mobile USA, L.P., and Wayport, Inc., November 15, 2016; (4) *Hitachi Maxell, LTD., v. Top Victory Electronics (Taiwan) Co. Ltd., et al.,* 2:14-cv-1121 JRG-RSP (E.D. Texas), on behalf of Top Victory Electronics (Taiwan) Co. LTD, *et. al.,* January 20, 2016; (5) *Sprint Spectrum, L.P. v. General Access Solutions, Ltd*., Petition for *Inter Partes Review* of U.S. Patent No. 7,173,916, on behalf of Sprint Spectrum L.P., July 13, 2018; (6) *Nichia Corporation v. Vizio, Inc*., 8:16-cv-00545; on behalf of Vizio, Inc., October 12, 2018; (7) *Intellectual Ventures I LLC, v. T-Mobile USA, Inc., T-Mobile US, Inc., Ericsson Inc., and Telefonaktiebolaget LM Ericsson*, 2:17-cv-00557 (JRG), on behalf of T-Mobile USA, Inc., T-Mobile US, Inc., Ericsson Inc., and Telefonaktiebolaget LM Ericsson, October 19, 2018; (8) Pfizer, Inc. v. Biogen, Inc., Petition for *Inter Partes Review* of U.S. Patent No. 8,821,873, on behalf of Pfizer, November 3, 2018; (9) *Finjan, Inc. v. ESET, LLC and ESET SPOL. S.R.O.*, 3:17-cv-00183-CAB-BGS, on behalf of ESET, January 15, 2019; (10) *Finjan, Inc. v. Cisco Systems, Inc.,* 5:17-cv-00072-BLF-SVK, on behalf of Cisco Systems, Inc., September 6, 2019; (11) *Facebook, Inc., Instagram, LLC and Whatsapp Inc. v. Blackberry Limited*, Petition for *Inter Partes Review* of U.S. Patent No. 9,349,120 B2, on behalf of Facebook, Inc., Instagram, LLC and Whatsapp Inc. December 20, 2019; (12) *3Shape A/S and 3Shape Inc. v. Align Technology, Inc.,* Petition for *Inter Partes Review* of U.S. Patent No. 7,156,661, IPR 2020-00222 and IPR

2020-00223, August 10, 2020, on behalf of 3Shape A/S and 3Shape Inc.; (13) *Finjan Inc. v. Rapid7, Inc. and Rapid7 LLC*, Northern District of Delaware; 1:18-cv-01519-MN, on behalf of Rapid7, Inc. and Rapid7 LLC, September 15, 2020; and, (14) *VLSI Technology LLC v. Intel Corporation*, Western District of Texas, 6:19-cv-00254, 6:19-cv-00255, 6:19-cv-00256, on behalf of Intel Corporation, September 23, 2020. I have not testified at trial.

11. My full curriculum vitae is attached hereto as Attachment 1.

III. PRELIMINARIES

A. Scope of Declaration and Legal Standards

12. I am not an attorney and will not offer opinions on the law. I am, however, rendering my expert opinion on the authenticity of the documents referenced herein and on when and how each of these documents was disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, could have located the documents before the dates discussed below with respect to the specific documents.

13. I am informed by counsel that a printed publication qualifies as publicly accessible as of the date it was disseminated or otherwise made available such that a person interested in and ordinarily skilled in the relevant subject matter could locate it through the exercise of ordinary diligence.

14. While I understand that the determination of public accessibility under the foregoing standard rests on a case-by-case analysis of the facts particular to an individual publication, I also understand that a printed publication is rendered "publicly accessible" if it is cataloged and indexed by a library such that a person interested in the relevant subject matter could locate it (i.e., I understand that cataloging and indexing by a library is sufficient, though there are other ways that a printed publication may qualify as publicly accessible). One manner of sufficient indexing is indexing according to subject matter category. I understand that the cataloging and indexing by a single library of a single instance of a particular printed publication is sufficient, even if the single library is in a foreign country. I understand that, even if access to a library is restricted, a printed publication that has been cataloged and indexed therein is publicly accessible so long as a presumption is raised that the portion of the public concerned with the relevant subject matter would know of the printed publication. I also understand that the cataloging and indexing of information that would guide a person interested in the relevant subject matter to the printed publication, such as the cataloging and indexing of an abstract for the printed publication, is sufficient to render the printed publication publicly accessible.

15. I understand that routine business practices, such as general library cataloging and indexing practices, can be used to establish an approximate date on

which a printed publication became publicly accessible.

B. Persons of Ordinary Skill in the Art

16. I am told by counsel that the subject matter of this proceeding generally relates to cryptography, and in particular a method for avoiding potential bias in the generation of a cryptographic key.

17. I have been informed by counsel that a "person of ordinary skill in the art at the time of the invention" (POSITA) is a hypothetical person who is presumed to be familiar with the relevant field and its literature at the time of the inventions. This hypothetical person is also a person of ordinary creativity, capable of understanding the scientific principles applicable to the pertinent field.

18. I am told by counsel that a person of ordinary skill in this subject matter or art as of December 27, 2000 would have possessed a bachelor's degree in computer science, mathematics, applied mathematics, or a related field, and (1) two or more years of experience in applied cryptography or computer security software engineering, and/or (2) an advanced degree in computer science or a related field. I have been further informed by counsel that a person of ordinary skill in the art would have been familiar with and able to understand the information known in the art relating to these fields, including the publications discussed in this declaration.

C. Use of Authoritative Databases

19. In preparing this report, I used authoritative databases, such as the OCLC bibliographic database, the Library of Congress Online Catalog, the *ACM Digital Library, Google Scholar, ResearchGate*, and *Semantic Scholar*, to confirm citation details of the publication discussed. Unless I note otherwise below in reference to a specific serial publication, it is my expert opinion that this standard protocol was followed for the publications discussed in Section V below.

20. *Indexing*. A researcher may discover material relevant to his or her topic in a variety of ways. One common means of discovery is to search for relevant information in an index of periodical and other publications. Having found relevant material, the researcher will then normally obtain it online, look for it in libraries, or purchase it from the publisher, a bookstore, a document delivery service, or other provider. Sometimes, the date of a document's public accessibility will involve both indexing and library date information.

21. Indexing services use a wide variety of controlled vocabularies to provide subject access and other means of discovering the content of documents. The formats in which these access terms are presented vary from service to service.

22. Online indexing services and digital repositories commonly provide bibliographic information, abstracts, and full-text copies of the indexed publications, along with a list of the documents cited in the indexed publication.

These services also often provide lists of publications that cite a given document. A citation of a document is evidence that the document was publicly available and in use by researchers no later than the publication date of the citing document.

23. *ACM Digital Library*.[1] This index is produced by the Association for Computing Machinery, the world's largest scientific and educational computing society. The *ACM Digital Library* contains the full text of all ACM publications, hosted full-text publications from selected publishers, and the ACM Guide to Computing Literature—a comprehensive bibliography of computing literature beginning in the 1950s with more than a million entries. All metadata in the database are freely available on the Web, including abstracts, linked references, citing work, and usage statistics. Full-text articles are available with subscription.

24. *ResearchGate*.[2] A social networking site for scientists and researchers to share papers, ask and answer questions, and find collaborators, *ResearchGate* is the largest academic social network in terms of active users, although other services have more registered users, and a 2015–2016 survey suggests that almost as many academics have Google Scholar profiles. Features available to *ResearchGate* members include following a research interest and the work of other individual participants, a blogging feature for users to write short

[1] https://dl.acm.org/
[2] www.researchgate.net

reviews on peer-reviewed articles, private chat rooms for sharing data, editing documents, or discussing confidential topics, and a research-focused job board. *ResearchGate* indexes self-published information on user profiles and suggests members to connect with others who have similar interests. Member questions are fielded to others who have identified relevant expertise on their profiles.

25. Founded in 2008, *ResearchGate* restricts its user accounts to people at recognized institutions and published researchers. Most of *ResearchGate's* users are involved in medicine, biology, engineering, computer science, agricultural sciences, and psychology. *ResearchGate* publishes a citation impact measurement in the form of an "RG Score," which is reported to be correlated with existing citation impact measures. *ResearchGate* does not charge fees for putting content on the site and does not require peer review.

26. *Semantic Scholar*.[3] Developed at the Allen Institute for Artificial Intelligence, Semantic Scholar was publicly released in November 2015. *Semantic Scholar* is an AI-backed search engine for scientific journal articles which uses a combination of machine learning, natural language processing, and machine vision to add a layer of semantic analysis to the traditional methods of citation analysis, and to extract relevant figures, entities, and venues from papers. *Semantic*

[3] www.semanticscholar.org

Scholar is designed to highlight important, influential papers, and to identify the connections between them.

27. As of January 2018, following a 2017 project that added biomedical papers and topic summaries, the *Semantic Scholar* corpus included more than 40 million papers from computer science and biomedicine. As of August 2019, the number of included papers had grown to more than 173 million after the addition of the Microsoft Academic Graph records, already used by Lens.org.

28. *Google Scholar*.[4] A freely accessible web search engine that indexes the full text or metadata of scholarly literature across an array of publishing formats and disciplines and released in beta in November 2004, the *Google Scholar* index includes a significant number of peer-reviewed online academic journals, books, conference papers, selected theses and dissertations, preprints, abstracts, technical reports, and other scholarly literature. Scientometric researchers estimate that *Google Scholar* contains roughly 389 million documents including articles, citations and patents making it the world's largest academic search engine in January 2018. The most relevant results for the searched keywords will be listed first, in order of the author's ranking, the number of references that are linked to it and their relevance to other scholarly literature, and the ranking of the publication that the journal appears in. Through its "cited by"

[4] https://scholar.google.com

feature, *Google Scholar* provides access to abstracts of articles that have cited the article being viewed.

29. *U.S. Copyright Office*. Created by Congress in 1897, the Copyright Office is responsible for administering a complex and dynamic set of laws, which include registration, the recordation of title and licenses, a number of statutory licensing provisions, and other aspects of the *1976 Copyright Act* and the *1998 Digital Millennium Copyright Act*. The public catalog in the Copyright Office includes information filed since 1978.[5] Individuals can search by title, personal or corporate name, key word, registration number, and document number. Works filed before 1978 can be located through the Copyright Public Records Reading Room.[6] A researcher can find the date on which an item was published and deposited for copyright.

IV. LIBRARY CATALOGING PRACTICES

A. MARC Records and OCLC

30. I am fully familiar with the library cataloging standard known as the MARC standard, which is an industry-wide standard method of storing and organizing library catalog information. MARC was first developed in the 1960's by the Library of Congress. A MARC-compatible library is one that has a catalog

[5] https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?DB=local&PAGE=First.
[6] https://www.copyright.gov/circs/circ23.pdf.

consisting of individual MARC records for works made available at that library.

31. Since at least the early 1970s and continuing to the present day, MARC has been the primary communications protocol for the transfer and storage of bibliographic metadata in libraries.[7] As explained by the Library of Congress:

> You could devise your own method of organizing the bibliographic information, but you would be isolating your library, limiting its options, and creating much more work for yourself. Using the MARC standard prevents duplication of work and allows libraries to better share bibliographic resources. Choosing to use MARC enables libraries to acquire cataloging data that is predictable and reliable. If a library were to develop a "home-grown" system that did not use MARC records, it would not be taking advantage of an industry-wide standard whose primary purpose is to foster communication of information.
>
> Using the MARC standard also enables libraries to make use of commercially available library automation systems to manage library operations. Many systems are available for libraries of all sizes and are designed to work with the MARC format. Systems are maintained and

[7] A complete history of the development of MARC can be found in *MARC: Its History and Implications* by Henrietta D. Avram (Washington, DC: Library of Congress, 1975) and available online from the Hathi Trust (https://babel.hathitrust.org/cgi/pt?id=mdp.39015034388556;view=1up;seq=1; last visited September 10, 2020).

> improved by the vendor so that libraries can benefit from the latest advances in computer technology. The MARC standard also allows libraries to replace one system with another with the assurance that their data will still be compatible.

Why Is a MARC Record Necessary? LIBRARY OF CONGRESS.[8]

32. Thus, almost every major library in the world is MARC-compatible. *See, e.g.*, *MARC Frequently Asked Questions (FAQ)*, LIBRARY OF CONGRESS.[9] ("MARC is the acronym for MAchine-Readable Cataloging. It defines a data format that emerged from a Library of Congress-led initiative that began nearly [fifty] years ago. It provides the mechanism by which computers exchange, use, and interpret bibliographic information, and its data elements make up the foundation of most library catalogs used today."). MARC is the ANSI/NISO Z39.2-1994 standard (reaffirmed in 2016) for Information Interchange Format. The full text of the standard is available from the Library of Congress.[10]

33. A MARC record comprises several fields, each of which contains specific data about the work. Each field is identified by a standardized, unique, three-digit code corresponding to the type of data that follow.[11] For example, a

---

[8] http://www.loc.gov/marc/umb/um01to06.html#part2
[9] https://www.loc.gov/marc/faq.html
[10] http://www.loc.gov/marc/bibliographic/
[11] http://www.loc.gov/marc/umb/um07to10.html; http://www.loc.gov/marc/bibliographic/

work's title is recorded in field 245, the primary author of the work is recorded in field 100, a work's International Standard Book Number ("ISBN") is recorded in field 020, a work's International Standard Serial Number ("ISSN") is recorded in field 022, and the publication date is recorded in field 260 under the subfield "c." [12] In some MARC records, field 264 is used rather than field 260 to record publication information.[13] Information in field 264 is similar to information in field 260 (Publication, Distribution, etc. (Imprint)). Field 264 is useful for cases where the content standard or institutional policies make a distinction between functions. If a work is a periodical, then its publication frequency is recorded in field 310, and the publication dates (e.g., the first and last publication) are recorded in field 362, which is also referred to as the enumeration/chronology field.[14, 15]

34. The library that initially created the MARC record is reflected in field

[12] http://www.loc.gov/marc/bibliographic/
[13] http://www.loc.gov/marc/bibliographic/bd264.html
[14] http://www.loc.gov/marc/bibliographic/bd3xx.html
[15] Upwards of two-thirds to three-quarters of book sales to libraries come from a jobber or wholesaler for online and print resources. These resellers make it their business to provide books to their customers as fast as possible, often providing turnaround times of only a single day after publication. Libraries purchase a significant portion of the balance of their books directly from publishers themselves, which provide delivery on a similarly expedited schedule. In general, libraries make these purchases throughout the year as the books are published and shelve the books as soon thereafter as possible in order to make the books available to their patrons. Thus, books are generally available at libraries across the country within just a few days of publication.

040 in subfield "a" with that library's unique library code.[16] Once a MARC record for a particular work is originally created by one library, other libraries can use that original MARC record to then create their own MARC records for their own copies of the same work. These other libraries may modify or add to the original MARC record as necessary to reflect data specific to their own copies of the work. However, the library that created the original MARC record would still be reflected in these modified MARC records (corresponding to other copies of the same work at other libraries) in field 040, subfield "a". The modifying library (or libraries) is reflected in field 040, subfield "d."[17]

35. I consulted the Directory of OCLC Libraries[18] in order to identify the institution that created or modified the MARC record. Moreover, when viewing the MARC record online via Online Computer Library Center's ("OCLC") bibliographic database, which I discuss further below, hovering over a library code in field 040 with the mouse reveals the full name of the library. I also used this method of "mousing over" the library codes in the OCLC database to identify the originating and modifying libraries for the MARC records discussed in this report.

36. MARC records also include one or more fields that show information

[16] http://www.loc.gov/marc/umb/um07to10.html; http://www.loc.gov/marc/bibliographic/
[17] http://www.loc.gov/marc/bibliographic/bd040.html
[18] http://www.oclc.org/contacts/libraries.en.html

regarding subject matter classification. For example, 6XX fields are termed "Subject Access Fields."[19] Among these, for example, is the 650 field; this is the "Subject Added Entry – Topical Term" field.[20] The 650 field is a "[s]ubject added entry in which the entry element is a topical term." *Id.* These entries "are assigned to a bibliographic record to provide access according to generally accepted thesaurus-building rules (e.g., *Library of Congress Subject Headings* (LCSH), *Medical Subject Headings* (MeSH))." *Id.*

37. Further, MARC records can include call numbers, which themselves contain a classification number. For example, a MARC record may identify a 050 field, which is the "Library of Congress Call Number."[21] A defined portion of the Library of Congress Call Number is the classification number, and "source of the classification number is *Library of Congress Classification* and the *LC Classification-Additions and Changes*." *Id.* Thus, the 050 field may be used to show information regarding subject matter classification. Further, the 082 field is the "Dewey Decimal Call Number."[22] A defined portion of the Dewey Decimal Call (DDC) Number is the classification number, and "source of the classification number is the *Dewey Decimal Classification and Relative Index*. Thus, included in

[19] http://www.loc.gov/marc/bibliographic/bd6xx.html
[20] http://www.loc.gov/marc/bibliographic/bd650.html
[21] http://www.loc.gov/marc/bibliographic/bd050.html
[22] http://www.loc.gov/marc/bibliographic/bd082.html.

the 082 field is a subject matter classification.

38. Each item in a library has a single classification number. A library selects a classification scheme (e.g., the Library of Congress Classification scheme just described or a similar scheme such as the Dewey Decimal Classification scheme) and uses it consistently. When the Library of Congress assigns the classification number, it appears as part of the 050 field, as discussed above. For MARC records created by libraries other than the Library of Congress (e.g., a university library or a local public library), the classification number may appear in a 09X (e.g., 090) field.[23]

39. When a MARC-compatible library acquires a work, it creates a MARC record for its copy of the work in its computer catalog system in the ordinary course of its business. This MARC record (for the copy of a work available at the particular library) may be later accessed by researchers in a number of ways. For example, many libraries, including the Library of Congress, make their MARC records available through their website. As an example, the MARC record for the copy of *The Unlikely Spy*, by Daniel Silva,[24] available at the Library of Congress can be viewed through the Library of Congress website, at https://catalog.loc.gov/vwebv/staffView?searchId=20265&recPointer=1&recCount

[23] http://www.loc.gov/marc/bibliographic/bd09x.html
[24] *The Unlikely Spy* is a 1996 novel written by Daniel Silva, who happens to be one of my favorite authors.

=25&bibId=2579985 (last visited September 10, 2020). One could, of course, always physically visit the library at which the work is available, and request to see that library's MARC record for the work. Moreover, members of the Online Computer Library Center ("OCLC") can access the MARC records of other member institutions through OCLC's online bibliographic database, as I explain further below.

B. Fields 008, 005, and 955 in MARC Records as Indicators of Public Accessibility

40. When a MARC-compatible library creates an original MARC record for a work, the library records the date of creation of that MARC record in **field 008**, characters 00 through 05, in the ordinary course of its business. *See* http://www.loc.gov/marc/bibliographic/bd008a.html (last visited September 10, 2020). For institutions that use OCLC software to create original MARC records, the date of creation in field 008 is automatically supplied by the OCLC software. The MARC record creation date in field 008 thus reflects the date on which, or shortly after which, a work was first acquired and cataloged by the library that created the original MARC record.

41. When other MARC-compatible libraries subsequently acquire their own copies of the same work, as mentioned, they create MARC records in their

own computer catalog systems for their copies in the ordinary course of business.[25] They may use a MARC record previously created for that work (by another MARC-compatible library) to create their own MARC records for their own copies of that same work.[26] The previously created MARC record used by OCLC participating libraries to create MARC records for their own copies may be obtained through the OCLC bibliographic database, as described above. If, when creating a MARC record to represent its own copy of the work, the OCLC participating library uses the master MARC record in its original form, the data in field 008 cannot be reentered or modified; therefore, the date in the 008 field will continue to reflect the date the MARC record was initially created by the originating library. On the other hand, if the OCLC participating library modifies the previously created MARC record when preparing its own MARC record for the local online catalog, data may be entered into the 008 field of its local MARC record.[27] But the library that created the original MARC record used by the OCLC

---

[25] Initial contributions to the OCLC bibliographic database for a work are called "master records."

[26] When a local library uses a master record in OCLC and produces (or downloads) it to the in-house system, the three-character symbol for the subsequent library is added to the holdings for the work.

[27] This practice is not required by, but is nevertheless consistent with, the MARC standard. Many MARC records exist today whose 008 fields indicate when the first original MARC record for a work was created, rather than when a derivative record was created based on the original MARC record by a subsequently acquiring library for its own computer catalog system.

participating library would still be reflected in the MARC record of the subsequently-acquiring library in field 040, subfield "a." Thus, the work identified by any MARC record possessed by any MARC-compatible library would have been accessible to the public at least as of the date shown in the 008 field, or shortly thereafter, either from the library that possesses the MARC record itself, or from the originating library indicated in field 040, subfield "a." As discussed, a MARC-compatible library in the ordinary course of its business creates a MARC record in its own catalog system for a work when it acquires a copy of that work.

42. Moreover, when a MARC record is created by a library for its own copy of a work, **field 005** is automatically populated with the date that MARC record was created in year, month, day format (YYYYMMDD). *See* http://www.loc.gov/marc/bibliographic/bd005.html (last visited September 10, 2020).[28] Thereafter, the library's computer system may automatically update the date in field 005 every time the library updates the MARC record (*e.g.*, to reflect that an item has been moved to a different shelving location within the library). *Id.* Thus, the work identified by any MARC record possessed by any MARC-compatible library would have been accessible to the public at least as of the date shown in the 005 field, or shortly thereafter, from the library that possesses the

[28] Some of the newer library catalog systems also include hour, minute, second (HHMMSS).

MARC record itself. As noted, because the 005 field may be updated each time the library updates its MARC record, the work identified by the MARC record may, in fact, have been accessible to the public from that library much earlier than the date indicated in the 005 field.

43. Moreover, MARC records for copies of works available at the Library of Congress can have a **955 field**. *See* http://www.loc.gov/cds/PDFdownloads/dcm/DCM_2007-03.pdf (last visited September 10, 2020).

44. Based on my personal experience as a professional librarian using the MARC and OCLC resources, it has been my experience that both resources were continuously operational and available since at least 1992. Indeed, in the course of my work, I have extensively used both resources over the past 30+ years, and I have consistently found the information contained within these resources to be complete and reliable. I have never found the date of accessibility as indicated in fields 008, 005, or 955 to be incorrect. And in only a minute number of cases have I found any errors at all in these records – none of which affected my ability to render an accurate opinion as to accessibility, indexing, or subject headings.

C. OCLC

45. The OCLC was created "to establish, maintain, and operate a computerized library network and to promote the evolution of library use, of

libraries themselves, and of librarianship, and to provide processes and products for the benefit of library users and libraries, including such objectives as increasing availability of library resources to individual library patrons and reducing the rate of rise of library per-unit costs, all for the fundamental public purpose of furthering ease of access to and use of the ever-expanding body of worldwide scientific, literary, and educational knowledge and information."[29] Among other services, OCLC and its members are responsible for maintaining the WorldCat database,[30] used by independent and institutional libraries throughout the world. All libraries that are members of OCLC are MARC-compatible.[31] OCLC-MARC records describes records produced since November 1993.[32] Like the two superseded OCLC documents, this revised set of guidelines is intended to assist catalogers in creating records for electronic resources in WorldCat, the OCLC Online Union Catalog. These guidelines pertain to OCLC-MARC tagging (that is, content designation). Cataloging rules and manuals (such as AACR2) govern the content

---

[29] Third Article, Amended Articles of Incorporation of OCLC Online Computer Library Center, Incorporated (available at https://www.oclc.org/content/dam/oclc/membership/articles-of-incorporation.pdf; last visited September 10, 2020).

[30] http://www.worldcat.org/

[31] https://help.oclc.org/Metadata_Services/OCLC-MARC_records/About_OCLC-MARC_records

[32] https://www.oclc.org/support/services/worldcat/documentation/cataloging/electronicresources.en.html

of records. You should implement these guidelines immediately.

46. When an OCLC member institution acquires a publication, like the other MARC-compatible libraries discussed above, it creates a MARC record for this publication in its computer catalog system in the ordinary course of its business. MARC records created at the Library of Congress are tape-loaded into the OCLC database through a subscription to MARC Distribution Services daily or weekly. Once the MARC record is created by a cataloger at an OCLC member library or is tape-loaded from the Library of Congress, the MARC record is then made available to any other OCLC members online, and thereby made available to the public. Accordingly, once the MARC record is created by a cataloger at an OCLC member library or is tape-loaded from the Library of Congress, any publication corresponding to the MARC record has been cataloged and indexed according to its subject matter such that a person interested in that subject matter could, with reasonable diligence, locate and access the publication through any library with access to the OCLC bibliographic database or through the Library of Congress.

47. When a MARC-compatible library creates an original MARC record for a work, the library records the date of creation of that MARC record in field 008, characters 00 through 05, in the ordinary course of its business.[33] For OCLC

[33] http://www.loc.gov/marc/bibliographic/bd008a.html

member institutions that use OCLC software to create original MARC records, the date of creation in field 008 is automatically supplied by the OCLC software. The MARC record creation date in field 008 thus reflects the date on which, or shortly after which, a work was first acquired and cataloged by the library that created the original MARC record.

48. When other MARC-compatible libraries subsequently acquire their own copies of the same work, as mentioned, they create MARC records in their own computer catalog systems for their copies in the ordinary course of business.[34] They may use a MARC record previously created for that work (by another MARC-compatible library) to create their own MARC records for their own copies of that same work.[35] The previously created MARC record used by subsequently-acquiring libraries to create MARC records for their own copies may be obtained through the OCLC bibliographic database, as described above. If, when creating a MARC record to represent its own copy of the work, the subsequently-acquiring library uses the master MARC record in its original form, the subsequently-acquiring library cannot reenter data into the 008 field; therefore, the date in the 008 field will continue to reflect the date the MARC record was initially created by

[34] Initial contributions to the bibliographic database for a work are called "master records."

[35] When a local library uses a master record in OCLC and produces (or downloads) it to the in-house system, the three-character symbol for the subsequent library is added to the holdings for the work.

the originating library. On the other hand, if the subsequently-acquiring library modifies the previously created MARC record when creating its own MARC record for its own copy of the work, the subsequently-acquiring library may enter into the 008 field of its own MARC record the date its own MARC record was created.[36] But the library that created the original MARC record used by the subsequently-acquiring library would still be reflected in the MARC record of the subsequently-acquiring library in field 040, subfield "a". Thus, the work identified by any MARC record possessed by any MARC-compatible library would have been accessible to the public at least as of the date shown in the 008 field, or shortly thereafter, either from the library that possesses the MARC record itself, or from the originating library indicated in field 040, subfield "a". As discussed, a MARC-compatible library in the ordinary course of its business creates a MARC record in its own catalog system for a work when it acquires a copy of that work.

49. Moreover, when a MARC record is created by a library for its own copy of a work, field 005 is automatically populated with the date that MARC record was created in year, month, day format (YYYYMMDD).[37] Thereafter, the

[36] This practice is not required by, but is nevertheless consistent with, the MARC standard. Many MARC records exist today whose 008 fields indicate when the first original MARC record for a work was created, rather than when a derivative record was created based on the original MARC record by a subsequently-acquiring library for its own computer catalog system.

[37] http://www.loc.gov/marc/bibliographic/bd005.html

library's computer system may automatically update the date in field 005 every time the library updates the MARC record (e.g., to reflect that an item has been moved to a different shelving location within the library). *Id.*[38] Thus, the work identified by any MARC record possessed by any MARC-compatible library would have been accessible to the public at least as of the date shown in the 005 field, or shortly thereafter, from the library that possesses the MARC record itself. As noted, because the 005 field may be updated each time the library updates its MARC record, the work identified by the MARC record may, in fact, have been accessible to the public from that library much earlier than the date indicated in the 005 field.

50. Moreover, MARC records for copies of works available at the Library of Congress can have a 955 field.[39] The 955 field in MARC records obtained from the Library of Congress provides Local Tracking Information, which is a record of internal steps in the cataloging process followed by the Library of Congress. *Id.* Entries in the 955 field for a particular work are generated by Library of Congress staff as the work progresses through the cataloging process. *Id.* One of the mandatory fields that library staff must enter for each step is the date (in the form

[38] Field 005 is visible when viewing a MARC record via an appropriate computerized interface. But when a MARC record is printed directly to hardcopy from the OCLC database, the "005" label is not shown. The date in the 005 field instead appears next to the label "Replaced."

[39] http://www.loc.gov/cds/PDFdownloads/dcm/DCM_2007-03.pdf

of "yyyy-mm-dd" or "yy-mm-dd"). *Id.* Thus, the work identified by a MARC record possessed by the Library of Congress would have been accessible to the public at least as of the earliest date shown in the 955 field, or shortly thereafter, from the Library of Congress.

51. Based on my personal experience as a professional librarian using the MARC and OCLC resources, it has been my experience that both of these resources were continuously operational and available since at least 1992. Indeed, in the course of my work, I have extensively used both of these resources over the past 30+ years, and I have consistently found the information contained within these resources to be complete and reliable. I have never found the date of accessibility as indicated in fields 008, 005, or 955 to be incorrect. And in only a minute number of cases have I found any errors at all in these records – none of which affected my ability to render an accurate opinion as to accessibility, indexing, or subject headings.

V. PUBLICATIONS IN THIS PROCEEDING

A. Publication 1 – Exhibit 1001: "'Pseudo-random' Number Generation Within Cryptographic Algorithms: The DDS Case" by Mihir Bellare, Shafi Goldwasser, and Daniele Micciancio (Bellare)[40]

52. Attached hereto as Exhibit 1001 is a true and correct copy of the

[40] I understand that the Bellare reference attached to my Declaration as Exhibit 1001 is numbered 1006 in the Petitioner's IPR petition.

conference paper "'Pseudo-random' Number Generation Within Cryptographic Algorithms: The DDS Case" by Mihir Bellare, Shafi Goldwasser, and Daniele Micciancio (hereafter "Bellare"). This conference paper was published in volume 1294 of the Lecture Notes in Computer Science series which is published as *Advances in Cryptology -- CRYPTO '97: 17th Annual International Cryptology Conference: Proceedings* on pages 277-291. The 17th Annual International Cryptology Conference (CRYPTO '97) was held August 17-21, 1997, in Santa Barbara, California. Exhibit 1001 is a true and correct copy of the conference paper within the conference proceedings book. I was able to find this book in the Linda Hall Library. The Bellare conference paper is also available digitally from the publisher, Springer, and in the digital repositories the *ACM Digital Library*[41] and *Semantic Scholar.*[42] Specifically, the text is complete; no pages are missing, and the text on each page appears to flow seamlessly from one page to the next; further, there are no visible alterations to the document. I was able to find the book within the custody of a library – a place where an authentic copy of this book would likely be. Exhibit 1001 is a true and correct copy in a condition that creates no suspicion about its authenticity.

[41] https://dl.acm.org/doi/10.5555/646762.706174
[42] https://www.semanticscholar.org/paper/%22Pseudo-Random%22-Number-Generation-Within-The-DDS-Bellare-Goldwasser/a6239b97df52667258b3714b001390dbddd0719b

53. Attached hereto as Attachment 1a is a true and correct copy of the MARC record for the *Advances in Cryptology -- CRYPTO '97: 17th Annual International Cryptology Conference: Proceedings* in the Linda Hall Library. The library ownership is indicated by the presence of the library's code (LHL) in the 049 field. I personally identified and retrieved the MARC record that is Attachment 1a.

54. The MARC record for the *Advances in Cryptology -- CRYPTO '97: 17th Annual International Cryptology Conference: Proceedings* shows that it was cataloged in the Linda Hall Library on September 2, 1997, as shown in field 008 ("970902"). The most recent enhancement to Attachment 1a occurred on October 20, 2014, as shown in field 005 ("20141020"). Therefore, this volume would have been available to users in the Linda Hall Library on or shortly after September 2, 1997.

55. Attached hereto as Attachment 1b is a true and correct copy of the MARC record for the *Advances in Cryptology -- CRYPTO '97: 17th Annual International Cryptology Conference: Proceedings* obtained from the OCLC bibliographic database. Attachment 1b indicates that a cataloger at the University of Iowa Library (Iowa City, Iowa) created OCLC record number 39161569 on September 16, 1997, as shown in the "Entered" field ("19970916"). The library continues to update this MARC record and enhanced the MARC record to meet

current cataloging rules. The most recent enhancement to Attachment 1b occurred on July 28, 2020, as shown in the "Replaced" field ("20200728"). I personally identified and retrieved the MARC record that is Attachment 1b.

56. Attachment 1b includes an entry in field 050 ("QA76.9.A25 $b C79 1997")—as described above, a subject matter classification number consistent with the Library of Congress classification system (analogous to the Dewey Decimal classification system) and an entry in field 082 ("005.82")—as described above, this includes a subject matter classification number consistent with the Dewey Decimal classification system. Attachment 1b further includes two English language descriptor terms reading "Computers $x Access control $v Congresses" (see Attachment 1c, Library of Congress subject heading sh2007008440) and "Cryptography $v Congresses" (see Attachment 1d, Library of Congress subject heading sh2008101157) in the 650 fields. Thus, as of its cataloging, the publication corresponding to the MARC record attached hereto as Attachment 1b was indexed according to its subject matter by virtue of at least three independently sufficient classifications: the field 050 entry, the field 082 entry, and the field 650 entries. As of September 16, 1997, the MARC record attached hereto as Attachment 1b was accessible through any library with access to the OCLC bibliographic database or the online catalog at a library that purchased this volume, which means that the corresponding publication was publicly available on or

before that same date through any library with access to the OCLC bibliographic database or through an individual library.

57. Attachment 1b indicates that the volume *Advances in Cryptology -- CRYPTO '97: 17th Annual International Cryptology Conference: Proceedings* as cataloged at the University of Iowa Library is currently available from 195 libraries (as shown in "195 other holdings"). In view of above, the volume *Advances in Cryptology -- CRYPTO '97: 17th Annual International Cryptology Conference: Proceedings* was publicly available on or shortly after September 16, 1997, because by that date it had been received, cataloged, and indexed at the University of Iowa Library and made part of the OCLC bibliographic database. For these reasons, it is my opinion that Exhibit 1001 was published and publicly accessible on or shortly after September 16, 1997.

58. Not only was Exhibit 1001 accessible and available to others in the field as of August 1997, researchers actually obtained and cited this conference paper in other works, which is still further confirmation of its availability and accessibility. For example, *Google Scholar* indicates that this conference paper has been cited 115 times (see Attachment 1e).

B. Publication 2 – Exhibit 1002: *LEDA: A Platform for Combinatorial and Geometric Computing* by Kurt Mehlhorn and Stefan Näher (Mehlhorn)[43]

59. Publication 2, attached as Exhibit 1002, is a copy of a book titled *LEDA: A Platform for Combinatorial and Geometric Computing* by Kurt Mehlhorn and Stefan Näher (hereafter "Mehlhorn") and issued by Cambridge University Press in 1999. Exhibit 1002 is a true and correct copy of the title page, title page verso, table of contents, and fourteen chapters as held by the Linda Hall Library. The Mehlhorn book is also available digitally and can be downloaded from the Internet.[44] Specifically, the texts of the fourteen chapters are complete; no pages are missing, and the text on each page appears to flow seamlessly from one page to the next; further, there are no visible alterations to the document. Exhibit 1002 was found within the custody of a library – a place where, if authentic, a copy of this book would likely be. Exhibit 1002 is a true and correct copy in a condition that creates no suspicion about its authenticity.

60. Attached hereto as Attachment 2a is a true and correct copy of the MARC record for this monograph from the Linda Hall Library online catalog. The library ownership is indicated by the presence of the library's code (LHL) in the 040 field. The most recent enhancement to Attachment 2a occurred on September

[43] I understand that the Mehlhorn reference attached to my Declaration as Exhibit 1002 is numbered 1007 in the Petitioner's IPR petition.
[44] https://people.mpi-inf.mpg.de/~mehlhorn/LEDAbook.html

3, 2011, as shown in field 005 ("2110903). I personally identified and retrieved the library catalog record which is Attachment 2a.

61. Based on finding a print copy of Exhibit 1002 in the Linda Hall Library and MARC record in its online library catalog attached as Attachment 2a, it is my opinion that the book *LEDA: A Platform for Combinatorial and Geometric Computing* by Mehlhorn and Näher was available in the Linda Hall Library as of March 12, 1999, as shown in field 008 ("990312"). The International Standard Book Number (ISBN) on Exhibit 1002 (0-5215-6329-1) matches the ISBN in field 020 of Attachment 2a. Therefore, Exhibit 1002 is the same book as the one that the cataloger at the Linda Hall Library used to create the MARC record that is Attachment 2a. Therefore, this volume would have been available to users in the Linda Hall Library on or shortly after March 12, 1999.

62. Attached hereto as Attachment 2b is a true and correct copy of the MARC record for the book *LEDA: A Platform for Combinatorial and Geometric Computing* by Mehlhorn and Näher obtained from the OCLC bibliographic database. I personally identified and retrieved the MARC record that is Attachment 2b. As previously noted, the library that created the record is recorded in field 040 with a unique library code. For Attachment 2b, that library code is "DLC," which means that the MARC record for this book was created at the Library of Congress. As can be seen in the "Entered" field in the MARC record

for this exhibit, a cataloger at the Library of Congress created OCLC record number 41017497 on March 12, 1999 and contributed it to the OCLC bibliographic database.

63. Attachment 2b further includes an entry in field 050 ("QA76.73.C153 $b M44 1999")—as described above, this includes a subject matter classification number consistent with the Library of Congress classification system (analogous to the Dewey Decimal classification system). Attachment 2b further includes an entry in field 082 ("005.13/3"), a subject matter consistent with the Dewey Decimal classification system. Attachment 2b further includes an English language field entry reading "C++ (Computer program language)" (see Attachment 2c, Library of Congress subject heading sh87007505) in the 650 field and a uniform title entry reading "LEDA (Computer program language)" (see Attachment 2d, Library of Congress subject heading sh94005928) in the 630 field. Thus, as of its cataloging, the publication corresponding to the MARC record attached hereto as Attachment 2b was indexed according to its subject matter by virtue of at least four independently sufficient classifications: the field 050 entry, the field 082 entry, the field 630 entry, and the field 650 entry. Further, as of March 12, 1999, the MARC record attached hereto as Attachment 2b was accessible through any library with access to the OCLC bibliographic database or the online catalog at a library that added this book to its collection, which means

that the corresponding publication was publicly available on or before that same date through any library with access to the OCLC bibliographic database or through an individual library.

64. Attachment 2b indicates that the book *LEDA: A Platform for Combinatorial and Geometric Computing* by Mehlhorn and Näher as cataloged at the Library of Congress is currently available from 196 libraries (as shown in "196 other holdings"). In view of above, this monograph *LEDA: A Platform for Combinatorial and Geometric Computing* was publicly available on or shortly after March 12, 1999, because by that date it had been cataloged and indexed at the Library of Congress, made part of the OCLC bibliographic database, and received at the Linda Hall Library. For these reasons, it is my opinion that Exhibit 1002 was published and accessible to the public on or shortly after March 12, 1999.

VI. CONCLUSION

65. In signing this declaration, I recognize that the declaration will be filed as evidence in a case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

66. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: September 29, 2020

Respectfully submitted,

Sylvia D. Hall-Ellis

Sylvia D. Hall-Ellis, Ph.D.

Attachment 1a

```
001    992820643405961
005    20141020201722.0
008    970902s1997    gw            000 0 eng d
010    ##$agb 97057103
015    ##$aGB97-57103
019    ##$a37558102
020    ##$a9783540633846
020    ##$a3540633847
035    ##$a(OCoLC)39161569
035    ##$9390974
035    ##$a(MoKL)282064-lhalldb
035    ##$a(lhalldb)282064-lhalldb
035    ##$z(OCoLC)37558102
040    ##$aOSU $cOSU $dUKM $dOCL $dLHL
049    ##$aLHLA
090    ##$aQA76.9.A25 $bC15 1997
111    2#$aCRYPTO (Conference) $n(17th : $d1997 : $cUniversity of
California, Santa Barbara)
245    10$aAdvances in cryptology, CRYPTO '97 : $b17th annual
international cryptology conference, Santa Barbara, California,
USA, August 17-21, 1997 : proceedings / $cBurton S. Kaliski Jr.,
(ed.)
246    30$aCRYPTO '97
260    ##$aBerlin ; $aNew York : $bSpringer, $cc1997.
300    ##$axii, 537 p. : $bill. ; $c24 cm.
490    1#$aLecture notes in computer science. Lecture notes in
artificial intelligence. $v1294
500    ##$a"Sponsored by the International Association for
Cryptologic Research (IACR), in cooperation with the IEEE Computer
Society Technical Committee on Security and Privacy and the
Computer Science Department of the University of California at
Santa Barbara (UCSB)"--Pref. p. vii.
504    ##$aIncludes bibliographical references and index.
650    #0$aComputers $xAccess control $vCongresses.
650    #0$aCryptography $vCongresses.
700    1#$aKaliski, Burton S.
710    2#$aInternational Association for Cryptologic Research.
710    2#$aIEEE Computer Society. $bTechnical Committee on Security
and Privacy.
710    2#$aUniversity of California, Santa Barbara. $bComputer
Science Department.
830    #0$aLecture notes in computer science. $pLecture notes in
artificial intelligence ; $v1294.
948    ##$aLTI 10/08/2013
```

Attachment 1b

OCLC 39161569 **No holdings in XXX - 195 other holdings**

Books

						Rec Stat		**Entered**	19970916	**Replaced**	20200728192455.0
Type	a	**ELvl**	M	**Srce**	d	**Audn**		**Ctrl**		**Lang**	eng
BLvl	m	**Form**		**Conf**	1	**Biog**		**MRec**		**Ctry**	gw
		Cont	b	**GPub**		**LitF**	0	**Indx**	1		
Desc	a	**Ills**	a	**Fest**	0	**DtSt**	s	**Dates**	1997,		

040 NUI $b eng $c NUI $d OSU $d UKM $d OCL $d UBA $d C$$Q $d OCLCQ $d NLGGC $d DRB $d ZWZ $d OCLCQ $d OHX $d BAKER $d DEBSZ $d TEU $d DEBBG $d OCLCQ $d OCLCF $d OCLCO $d OCLCQ $d OCL $d OCLCO $d OCLCA $d OCLCQ $d OCLCO $d UWO $d OCLCA $d OCLCQ $d UKBTH $d MM9
015 GB9757103 $2 bnb
015 97,N33,0366 $2 dnb
015 00 b9757103
016 7 950848476 $2 DE-101
016 7 011013943 $2 Uk
019 37558102 $a 40291912 $a 1170551680
020 3540633847 $q (alk. paper)
020 9783540633846 $q (alk. paper)
037 $c No price
050 4 QA76.9.A25 $b C79 1997
072 7 QA $2 lcco
080 518.24
082 04 005.82 $2 21
084 54.62 $2 bcl
084 SS 1997 $2 rvk
084 SS 4800 $2 rvk
084 ST 276 $2 rvk
084 28 $2 sdnb
084 DAT 465f $2 stub
090 $b
111 2 CRYPTO (Conference) $d (1997 : $c University of California, Santa Barbara)
245 10 Advances in cryptology--CRYPTO '97 : $b 17th annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997 : proceedings / $c Burton S. Kaliski Jr., (ed.).
246 30 CRYPTO '97
260 Berlin ; $a New York : $b Springer, $c ©1997.
300 xii, 537 pages : $b illustrations ; $c 24 cm
336 text $b txt $2 rdacontent
337 unmediated $b n $2 rdamedia
338 volume $b nc $2 rdacarrier
490 1 Lecture notes in computer science, $x 0302-9743 ; $v 1294
500 "Sponsored by the International Association for Cryptologic Research (IACR), in cooperation with the IEEE Computer Society Technical Committee on Security and Privacy, Computer Science Department of the University of California at Santa Barbara--P. vii.
504 Includes bibliographical references and index.

530 Also available via the World Wide Web.
520 Taken from the 17th Annual International Cryptology Conference, the papers in this volume focus on advances in cryptology. They are organized in sections, including complexity theory, cryptographic primitives, lattice-based cryptography, digital signatures, and information theory.
650 0 Computers $x Access control $v Congresses.
650 0 Cryptography $v Congresses.
650 7 Computers $x Access control. $2 fast $0 (OCoLC)fst00872779
650 7 Cryptography. $2 fast $0 (OCoLC)fst00884552
650 7 Kryptologie $2 gnd $0 (DE-588)4033329-2
650 7 Kongress $2 gnd $0 (DE-588)4130470-6
650 17 Geheimschrift. $2 gtt
650 17 Coderingstheorie. $2 gtt
650 17 Dataprocessing. $2 gtt
650 7 Computacao aplicada. $2 larpcal
650 7 Computabilidade e modelos de computacao. $2 larpcal
650 7 Cryptographie $x Congrès. $2 ram
650 7 Systèmes informatiques $x Mesures de sûreté $x Congrès. $2 ram
655 4 Kongress $z Santa Barbara (Calif.) $y 1997.
655 7 Conference papers and proceedings. $2 fast $0 (OCoLC)fst01423772
655 7 Santa Barbara (Calif., 1997) $2 swd
700 1 Kaliski, Burton S.
710 2 International Association for Cryptologic Research.
710 2 IEEE Computer Society. $b Technical Committee on Security and Privacy.
710 2 University of California, Santa Barbara. $b Computer Science Department.
830 0 Lecture notes in computer science ; $v 1294. $x 0302-9743
856 41 $3 Table of contents $u http://swbplus.bsz-bw.de/bsz060820756inh.htm
856 41 $u http://link.springer-ny.com/link/service/series/0558/tocs/t1294.htm $z Restricted to Springer LINK subscribers
856 4 $3 Kapitel 1 $u http://swbplus.bsz-bw.de/bsz060820756kap.htm
856 4 $u http://link.springer.de/link/service/series/0558/tocs/t1294.htm $z Sommaire + texte intégral (accès réservé)
856 4 $u http://link.springer-ny.com/link/service/series/0558/tocs/t1294.htm $z Connect to Internet resource

Attachment 1c

LIBRARY OF CONGRESS AUTHORITIES

Library buildings are closed to the public until further notice, but LC Authorities is available. More.

LC control no.: sh2007008440
LCCN Permalink: https://lccn.loc.gov/sh2007008440
HEADING: Computers Access control Congresses

000 00518cz a2200181n 450
001 7307419
005 20110729155639.0
008 070926 | anannbabn |n ana
010 __ |**a** sh2007008440
035 __ |**a** (DLC)7307419
035 __ |**a** (DLC)sh2007008440
035 __ |**a** (DLC)366684
040 __ |**a** DLC |**b** eng |**c** DLC
150 __ |**a** Computers |**x** Access control |**v** Congresses
667 __ |**a** Record generated for validation purposes.
670 __ |**a** Work cat.: Cryptology and computational number theory, c1990
906 __ |**t** 8888 |**u** te00 |**v** 0
953 __ |**a** tj03

◀ Previous Next ▶

Save, Print or Email Records (View Help)

Select Download Format: Text (Labelled Display) Press to SAVE or PRINT

Email Text (Labelled Display) to: Press to SEND EMAIL

Library of Congress
URL: https://www.loc.gov/

Mailing Address:
101 Independence Ave, S.E.
Washington, DC 20540

Library of Congress Authorities
URL: http://authorities.loc.gov/
Library of Congress Online Catalog
URL: https://catalog.loc.gov/

Questions, comments, error reports: Contact Us

Attachment 1d

LIBRARY OF CONGRESS AUTHORITIES

LC control no.: sh2008101157
LCCN Permalink: https://lccn.loc.gov/sh2008101157
HEADING: Cryptography Congresses

000 00459cz a2200169n 450
001 7437687
005 20110729160102.0
008 080206|| anannbabn |n ana
010 __ |**a** sh2008101157
035 __ |**a** (DLC)376693
035 __ |**a** (DLC)sh2008101157
040 __ |**a** DLC |**b** eng |**c** DLC
150 __ |**a** Cryptography |**v** Congresses
667 __ |**a** Record generated for validation purposes.
670 __ |**a** Work cat.: Cryptography and lattices, c2001
906 __ |**t** 8888 |**u** te00 |**v** 0
953 __ |**a** te00

◀ Previous Next ▶

Save, Print or Email Records (View Help)

Select Download Format: Text (Labelled Display) — Press to SAVE or PRINT

Email Text (Labelled Display) to: — Press to SEND EMAIL

Library of Congress
URL: https://www.loc.gov/

Mailing Address:
101 Independence Ave, S.E.
Washington, DC 20540

Library of Congress Authorities
URL: http://authorities.loc.gov/
Library of Congress Online Catalog
URL: https://catalog.loc.gov/

Questions, comments, error reports: Contact Us

Attachment 1e

“Pseudo-random” number generation within cryptographic algorithms: The DDS case

M Bellare, S Goldwasser, D Micciancio - Annual International Cryptology …, 1997 - Springer

The DSS signature algorithm requires the signer to generate a new random number with every signature. We show that if random numbers for DSS are generated using a linear congruential pseudorandom number generator (LCG) then the secret key can be quickly recovered after seeing a few signatures. This illustrates the high vulnerability of the DSS to weaknesses in the underlying random number generation process. It also confirms, that a sequence produced by LCG is not only predictable as has been known before, but should …

☆ Cited by 115 Related articles All 16 versions

Showing the best result for this search. See all results

Attachment 2a

```
001    993303633405961
005    20110903204957.0
008    990312s1999    enka     b    001 0 eng
010    ##$a   99024952
015    ##$aGB99-76929
019    ##$a42834752
020    ##$a0521563291 (hb)
035    ##$9452597
035    ##$a(OCoLC)41017497
035    ##$a(MoKL)330363-lhalldb
035    ##$a(lhalldb)330363-lhalldb
035    ##$z(OCoLC)42834752
040    ##$aDLC $cDLC $dC#P $dUKM $dLHL
049    ##$aLHLA
050    00$aQA76.73.C153 $bM44 1999
100    1#$aMehlhorn, Kurt, $d1949-
245    10$aLeda : $ba platform for combinatorial and geometric
computing / $cKurt Mehlhorn, Stefan Näher.
260    ##$aCambridge, U.K. ; $aNew York : $bCambridge University
Press, $c1999.
300    ##$axvi, 1018 p. : $bill. ; $c26 cm.
504    ##$aIncludes bibliographical references (p. 992-1001) and
index.
630    00$aLEDA (Computer file)
650    #0$aC++ (Computer program language)
700    1#$aNäher, Stefan
948    ##$aLTI 09/01/2008
994    ##$aE0 $bLHL
```

Attachment 2b

OCLC 41017497 **No holdings in XXX - 196 other holdings**

Books

						Rec Stat		**Entered**	19990312	**Replaced**	20200605170627.6
Type	a	**ELvl**	4	**Srce**		**Audn**		**Ctrl**		**Lang**	eng
BLvl	m	**Form**		**Conf**	0	**Biog**		**MRec**		**Ctry**	nyu
		Cont	b	**GPub**		**LitF**	0	**Indx**	0		
Desc	a	**Ills**	a	**Fest**	0	**DtSt**	s	**Dates**	1999 ,		

010 99024952
040 DLC $b eng $c DLC $d C#P $d UKM $d UBA $d BAKER $d NLGGC $d BTCTA $d YDXCP $d OCLCG $d IG# $d OCLCQ $d ZWZ $d OCLCQ $d OCLCF $d OCLCO $d BDX $d OCLCQ $d RDF $d NAM $d OCLCQ $d IOG $d GZM $d OCLCQ $d RAO $d OCLCO $d OCLCQ $d OCLCO $d MTU $d OCLCQ $d OCLCO $d OCLCQ $d SRU
015 GB9976929 $2 bnb
015 GB99-76929
019 42834752 $a 490712542 $a 1156793153
020 0521563291 $q (hb)
020 9780521563291 $q (hb)
042 pcc
050 00 QA76.73.C153 $b M44 1999
082 00 005.13/3 $2 21
084 54.52 $2 bcl
090 $b
100 1 Mehlhorn, Kurt, $d 1949-
245 10 Leda : $b a platform for combinatorial and geometric computing / $c Kurt Mehlhorn, Stefan Näher.
260 New York : $b Cambridge University Press, $c 1999.
300 xvi, 1018 pages : $b illustrations ; $c 26 cm
336 text $b txt $2 rdacontent
337 unmediated $b n $2 rdamedia
338 volume $b nc $2 rdacarrier
504 Includes bibliographical references (pages 992-1001).
505 0 Introduction -- 1. Foundations -- 2. Basic Data Types -- 3. Numbers and Matrices -- 4. Advanced Data Types -- 5. Graphs and their Data Structures -- 6. Graph Algorithms -- 7. Embedded Graphs -- 8. The Geometry Kernels -- 9. Geometry Algorithms -- 10. Windows and Panels -- 11. GraphWin -- 12. On the Implementation of LEDA -- 13. Manual Pages and Documentation -- 14. Bibliography.
520 1 "LEDA is a library of efficient data types and algorithms and a platform for combinatorial and geometric computing on which application programs can be built. This is the first book devoted to the library. Written by the main authors of LEDA it is the definitive account - describing how the system is constructed, how it operates and how it can be used. The authors supply ample examples from a range of areas to show how the library can be used in practice, making the book essential for all workers in algorithms, data structures and computational geometry."--Jacket.
630 00 LEDA (Computer file)
650 0 C++ (Computer program language)
630 07 LEDA (Computer file) $2 fast $0 (OCoLC)fst01391382

650 7 C++ (Computer program language) $2 fast $0 (OCoLC)fst00843286
650 17 Object-georiënteerd programmeren. $2 gtt
650 17 Algoritmen. $2 gtt
650 17 STL. $2 gtt
650 7 Linguagem de programacao (outras) $2 larpcal
650 7 C (langage de programmation) $2 ram
700 1 Näher, Stefan.
856 41 $3 Table of contents $u http://catdir.loc.gov/catdir/toc/cam027/99024952.html
856 42 $3 Publisher description $u http://catdir.loc.gov/catdir/description/cam0210/99024952.html
856 4 $u http://www.mpi-sb.mpg.de/~mehlhorn/LEDAbook.html $z site web de l'auteur

Attachment 2c

LC control no.: sh 87007505

LCCN Permalink: https://lccn.loc.gov/sh87007505

HEADING: C++ (Computer program language)

000 01496cz a2200265n 450

001 4812443

005 20120329073127.0

008 871202|| anannbabn |a ana

010 __ |**a** sh 87007505

035 __ |**a** (DLC)sh 87007505

035 __ |**a** (DLC)159128

035 __ |**a** (DLC)6978366

035 __ |**a** (DLC)sp 87007505

035 __ |**a** (DLC)349680

040 __ |**a** DLC |**c** DLC |**d** DLC |**d** ABAU |**d** DLC

053 _0 |**a** QA76.73.C153

150 __ |**a** C++ (Computer program language)

450 __ |**a** C Plus Plus (Computer program language)

550 __ |**w** g |**a** Object-oriented programming languages

670 __ |**a** Work cat.: 85-20087: Stroustrup, B. The C++ programming language, 1986 |**b** (C++ is a general purpose programming language designed to make programming more enoyable for the serious programmer. Except for minor details, C++ is a superset of the C programming language. In addition to the facilities provided by C, C++ provides flexible and efficient facilities for defining new types)

670 __ |**a** Wiener, R. An introduction to object-oriented programming and C++, 1988 |**b** (C++ is a hybrid language that provides a fusion of object-oriented functionality with the features of a traditional and efficient structured language, C. C++, developed at AT&T Bell Laboratories in the early 1980's ... is a superset of C that includes support for object-oriented programming)

670 __ |**a** Alltheweb search, Mar. 12, 2004 (C Plus Plus)

906 __ |**t** 0899 |**u** te04 |**v** 0

952 __ |**a** LC pattern: C (Computer program language)

953 __ |**a** fg03 |**b** ta29

Save, Print or Email Records (View Help)
Select Download Format: Text (Labelled Display) Press to SAVE or PRINT
Email Text (Labelled Display) to: Press to SEND EMAIL

Help - Search - Search History - Headings List - Start Over

Library of Congress
URL: https://www.loc.gov/

Mailing Address:
101 Independence Ave, S.E.
Washington, DC 20540

Library of Congress Authorities
URL: http://authorities.loc.gov/
Library of Congress Online Catalog
URL: https://catalog.loc.gov/

Questions, comments, error reports: Contact Us

Attachment 2d

LIBRARY OF CONGRESS AUTHORITIES

LC control no.: sh 94005928
LCCN Permalink: https://lccn.loc.gov/sh94005928
HEADING: Leda (Computer program language)

000 00607cz a2200193n 450
001 4861680
005 20120330081808.0
008 940817|| anannbabn |a ana
010 __ |**a** sh 94005928
035 __ |**a** (DLC)sh 94005928
035 __ |**a** (DLC)208365
040 __ |**a** DLC |**c** DLC
150 __ |**a** Leda (Computer program language)
550 __ |**w** g |**a** Programming languages (Electronic computers)
670 __ |**a** Work cat.: 94-34433: Budd, T. Multiparadigm programming in Leda, c1995.
906 __ |**t** 9436 |**u** te03 |**v** 1
952 __ |**a** 0 bib. record(s) to be changed
952 __ |**a** LC pattern: StarLogo (Computer program language)
953 __ |**a** jf04

◀ Previous Next ▶

Save, Print or Email Records (View Help)	
Select Download Format: Text (Labelled Display)	Press to SAVE or PRINT
Email Text (Labelled Display) to:	
	Press to SEND EMAIL

Library of Congress
URL: https://www.loc.gov/

Mailing Address:
101 Independence Ave, S.E.
Washington, DC 20540

Library of Congress Authorities
URL: http://authorities.loc.gov/
Library of Congress Online Catalog
URL: https://catalog.loc.gov/

Questions, comments, error reports: Contact Us

Exhibit 1001

Burton S. Kaliski Jr. (Ed.)

Advances in Cryptology – CRYPTO '97

17th Annual International
Cryptology Conference
Santa Barbara, California, USA
August 17-21, 1997
Proceedings

Springer

Volume Editor

Burton S. Kaliski Jr.
RSA Laboratories
20 Crosby Drive, Bedford, MA 01730-1402, USA
E-mail: burt@rsa.com

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Advances in cryptology : proceedings / CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17 - 21, 1997. Burton S. Kaliski (ed.). [IACR]. - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Santa Clara ; Singapore ; Tokyo : Springer, 1997
(Lecture notes in computer science ; Vol. 1294)
ISBN 3-540-63384-7

CR Subject Classification (1991): E.3, G.2.1, D.4.6, K.6.5,F.2.1-2, C.2, J.1, E.4

ISSN 0302-9743
ISBN 3-540-63384-7 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10546375 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

Preface

Crypto '97, the Seventeenth Annual Crypto conference organized by the International Association for Cryptologic Research (IACR) in cooperation with the IEEE Computer Society Technical Committee on Security and Privacy and the Computer Science Department of the University of California, Santa Barbara, represents another step forward in the steady progression of the science of cryptology. There is both a tremendous need for and a great amount of work on securing information with cryptologic technology. As one of the two annual meetings held by the IACR, the Crypto conference provides a focal point for presentation and discussion of research on all aspects of this science.

It is thus a privilege to coordinate the efforts of this community in focusing on its steps forward. Crypto '97 is a conference for its community, and to the researchers who have contributed to it — those whose papers appear in the proceedings, those whose submissions were not accepted, and those who have laid the foundation for the work — the community owes a debt of gratitude.

The process of developing a conference program is a challenging one, and this year's committee made the process both enjoyable and effective. My thanks go to Antoon Bosselaers, Gilles Brassard, Johannes Buchmann, Ivan Damgård, Donald Davies, Alfredo de Santis, Susan Langford, James L. Massey, Moni Naor, David Naccache, Tatsuaki Okamoto, Douglas Stinson, Michael J. Wiener, Rebecca Wright, and Yuliang Zheng for many hours of reviewing submissions and presenting their comments to the committee.

My thanks also to the committee's two advisory members, Neal Koblitz and Hugo Krawcyzk, the program chairs of Crypto '96 and '98. Neal's experience from a year ago and Hugo's perspective on the year ahead have helped to make this year's conference what it is, and should provide continuity to the next one.

Continuing a recent tradition, the review process for Crypto '97 was conducted entirely by e-mail and fax, without a program committee meeting. Each submission was assigned anonymously to three committee members (though many submissions were reviewed by more than three people), and decisions were made through several rounds of e-mail discussions. Of the 160 submissions received, the committee accepted 36, of which 35 appear in final form in these proceedings. Except for the papers themselves, nearly all correspondence with authors was also conducted by e-mail.

CRYPTO '97

August 17–21, 1997, Santa Barbara, California, USA

Sponsored by the

International Association for Cryptologic Research (IACR)

in cooperation with

IEEE Computer Society Technical Committee on Security and Privacy
Computer Science Department, University of California, Santa Barbara

General Chair

Bruce Schneier, Counterpane Systems, USA

Program Chair

Burt Kaliski, RSA Laboratories, USA

Program Committee

Antoon Bosselaers ..... Katholieke Universiteit Leuven, Belgium
Gilles Brassard ..... Université de Montréal, Canada
Johannes Buchmann ..... Technische Hochschule Darmstadt, Germany
Ivan Damgård ..... Aarhus University, Denmark
Donald Davies ..... Royal Holloway College London, United Kingdom
Alfredo de Santis ..... Università di Salerno, Italy
Susan Langford ..... Atalla Corporation, USA
James L. Massey ..... Swiss Federal Institute of Technology, Switzerland
Moni Naor ..... Weizmann Institute, Israel
David Naccache ..... Gemplus, France
Tatsuaki Okamoto ..... NTT Laboratories, Japan
Douglas Stinson ..... University of Nebraska, USA
Michael J. Wiener ..... Entrust Technologies, Canada
Rebecca Wright ..... AT&T Labs, USA
Yuliang Zheng ..... Monash University, Australia

Advisory Members

Neal Koblitz *(Crypto '96 program chair)* ..... University of Washington, USA
Hugo Krawczyk *(Crypto '98 program chair)* IBM T.J. Watson Research Center, USA
..... and Technion, Israel

Contents

Cryptanalysis of Public-Key Cryptosystems (I)

Cryptanalysis of Public-Key Cryptosystems (II)

Information Theory

Invited Lecture

Elliptic Curve Implementation

Number-Theoretic Systems

Distributed Cryptography

Hash Functions

Cryptanalysis of Secret-Key Cryptosystems

"Pseudo-Random" Number Generation Within Cryptographic Algorithms: The DDS Case

Mihir Bellare[1], Shafi Goldwasser[2] and Daniele Micciancio[2]

[1] Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-Mail: mihir@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/mihir.

[2] MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139, USA. E-Mail: {shafi,miccianc}@theory.lcs.mit.edu.

Abstract. The DSS signature algorithm requires the signer to generate a new random number with every signature. We show that if random numbers for DSS are generated using a linear congruential pseudorandom number generator (LCG) then the secret key can be quickly recovered after seeing a few signatures. This illustrates the high vulnerability of the DSS to weaknesses in the underlying random number generation process. It also confirms, that a sequence produced by LCG is not only predictable as has been known before, but should be used with extreme caution even within cryptographic applications that would appear to protect this sequence. The attack we present applies to truncated linear congruential generators as well, and can be extended to any pseudo random generator that can be described via modular linear equations.

1 Introduction

Randomness is a key ingredient for cryptography. Random bits are necessary not only for generating cryptographic keys, but are also often an integral part of steps of cryptographic algorithms. Examples are the DSS signature algorithm [16] which requires the choice of a new random number every time a new signature is generated, and CBC encryption, which requires the generation of a new random IV each time a new message is encrypted. (In fact, any secure, stateless encryption scheme must be probabilistic, requiring new randomness for each encryption [8].) In some cases, the random numbers chosen may have to be kept secret (as for DSS, where the leakage of one such random number compromises the secret key), whereas for other cases they can be made public (as in CBC encryption, where the IV may be sent in the clear).

In practice, the random bits will be generated by a pseudo random number generation process. For example, the DSS description [16] explicitly allows either using random or pseudo-random numbers. When this is done, the security of the scheme of course depends in a crucial way on the quality of the random bits produced by the generator. Thus, an evaluation of the overall security of a cryptographic algorithm should consider and take into account the choice of the pseudorandom generator.

It has been well accepted that a good notion of pseudorandomness for cryptographic purposes is ***unpredictability*** [18,20,3,7]: given an initial sequence produced by a pseudo-random number generator on an unknown seed, it is hard to predict with better probability than guessing at random, the next bit in the sequence output by the generator. Such generators can be constructed based on number-theoretic assumptions, but are computationally costly. Alternatively, one could build a generator out of DES which would be unpredictable assuming DES behaves like a pseudorandom function, but in some contexts this may be deemed costly too, or we might not want to make such a strong assumption. Since using a weaker generator does not necessarily mean the resulting cryptographic algorithm is insecure, in practice one usually uses some weak but fast generator.

The intent of our paper is to illustrate the extreme care with which one should choose a pseudo random number generator to use within a particular cryptographic algorithm. Specifically, we consider a concrete algorithm, the Digital Signature Standard [16], and a concrete pseudo random number generator, the linear congruential generator (LCG) or truncated linear congruential pseudo random generator. We then show that if a LCG or truncated LCG is used to produce the pseudo random choices called for in DSS, then DSS becomes completely breakable.

We remark that the Standard [16] recommends the use of a pseudo-random generator based on SHA-1 or DES. The attack we describe does not say anything about the use of DSS with such generators, but it does illustrates the high vulnerability of the DSS to the underlying random number generation process.

We remark that LCGs are known to be predictable if part of the pseudo-random sequence is made public (see section 1.2 for details). However in DSS none of the pseudo-random numbers used is ever revealed, and thus predictability does not imply insecurity here.

Let us now look at all this more closely.

1.1 Pseudorandom numbers in DSS

Recall that the DSS has public parameters p, q, g where p, q are primes, of 512 bits and 160 bits respectively, and g is a generator of an order q subgroup of Z_p^*. The signer has a public key $y = g^x$ where $x \in Z_q$. To sign a message $m \in Z_q$, the signer picks at random a number $k \in \{1, \ldots, q-1\}$ and computes a signature (r, s), where $r = (g^k \bmod p) \bmod q$ and $s = (xr + m)k^{-1} \bmod q$.

Here the "nonce" k is chosen at random, anew for each message. In practice, a sequence of nonces will be produced by a generator $\mathcal{G}$ which, given some initial seed k_0, produces a sequence of values $k_1, k_2, \ldots$; k_i will be the nonce for the i-th signature.

The adversary (cryptanalyst) sees the public key y, and triples (m_i, r_i, s_i) where (r_i, s_i) is a signature of m_i. Notice that the secrecy of the nonces is crucial. If ever a single nonce k_i is revealed to the adversary, then the latter can recover the secret key x, because $x = (s_i k_i - m_i) r_i^{-1} \bmod q$. However, the nonces appear to be very well protected, making it hard to exploit any such weakness. The

cryptanalyst only sees $r_i = (g^{k_i} \bmod p) \bmod q$ from which he cannot recover k_i short of computing discrete logarithms, and in fact not even then, due to the second mod operation. So even if $\mathcal{G}$ is a predictable generator, meaning, say, that given k_1, k_2 we can find k_3, there is no a priori reason to think DSS is vulnerable with this generator, because how can the cryptanalyst ever get to know k_1, k_2 anyway?

This might encourage a user to think that even a weak (predictable) generator is OK for DSS. This view would be wrong. We indicate that in fact DSS is vulnerable, because without a sufficiently good pseudorandom number generation process, the "masking" of the nonces provided by the algorithm is not sufficient to protect the nonces, even though recovering them seems a priori to require solving the discrete logarithm problem. In fact we prove a quite general lemma showing why this masking is essentially ineffective for pretty much *any* pseudorandom generator, and show specifically how to recover the keys when the generator is an LCG or truncated LCG. Thus one should not succumb to the temptation of using a weak generator for DSS.

1.2 Linear congruential generators

Recall that linear congruential generators are pseudo-random number generators based on a linear recurrence $X_{n+1} = aX_n + b \bmod M$ where a, b and M are parameters initially chosen at random and then fixed, and the seed is the initial value X_0. The advantage of linear congruential generators is that they are fast, and it has been shown [11] that they have good statistical properties for appropriate choices of the parameters a, b, M.

On the other hand, their unpredictability properties are known to be quite weak. Clearly they are predictable in their simplest form: if the parameters a, b and M are known, given X_0 all the other X_n can be easily computed. Plumstead (Boyar) [17] shows that even if the parameters a, b, M are unknown the sequence of numbers produced by a linear congruential generator is still predictable given some of the X_i. Truncated LCG were suggested by Knuth [12] as a possible way to make a linear congruential generator secure. However these generators have also been shown to be predictable [5,9,19] as have more general congruential generators [4,13].

However, as indicated above, this predictability does not directly mean a cryptographic algorithm using the generator is breakable, since it is possible none of the bits of the random numbers used by the algorithm are ever made public. DSS is (was) a case in point.

1.3 Cryptanalysis of DSS with LCG

DSS WITH LCG. We consider what happens when the nonces in DSS are generated using an LCG with known parameters a, b, M and hidden seed k_0. The predictability of the generator does not a priori appear to be a problem, due to the masking provided by the algorithm as indicated above. However, given just three valid signatures, we show how to recover the secret key.

UNIQUENESS LEMMA. We begin with a general lemma which indicates why the above intuition that the DSS protects the nonces may be false. The lemma (called

the Uniqueness Lemma) says that as long as the nonces are pseudorandomly generated then, even if we *ignore* the relations $r_i = (g^{k_i} \bmod p) \bmod q$, the DSS signature equations $s_i k_i - r_i x = m_i$ uniquely determine the secret key with high probability. This means the cryptanalyst can effectively ignore the masking that is supposed to protect the nonces. This is true for *any* pseudorandom generation process, even a cryptographically strong one, using an unpredictable generator. This lemma tells us we can concentrate on the signature equations.

SOLVING THE EQUATIONS. We begin the cryptanalysis of DSS with LCG by combining the DSS "signature equations" with the LCG generation equations to get a system of equations. (In the process we ignore the $r_i = (g^{k_i} \bmod p) \bmod q$ relations, invoking the Uniqueness Lemma to say that solving the signature equations suffices to find the secret key.) However, this system is not trivial to solve because it is a system of simultaneous *modular* equations in *different* moduli. Techniques like Gaussian elimination fail. Instead we turn to lattice reduction. We show how to use Babai's closest vector approximation algorithm to solve such a system. The main difficulty here is dealing with the fact that this algorithm only returns (not very good) approximations to the closest vector. We then extend this to the case of the truncated LCG.

1.4 Other results, discussion, and implications

We extend our techniques to provide a general algorithm for solving a system of simultaneous linear modular equations in different moduli. (Another way of doing this, when the number of equations is constant, is to reduce the problem to integer programming in constant dimension and apply the algorithms of [14,10]. Our alternative solution seems simpler and more direct.)

In many cryptographic algorithms, the random numbers used are processed in a way that the public information gives little information about the original numbers. This is the case for the nonces in DSS. In such a setting, it may be reasonable to think that weak random number generators can suffice: even predictable generators could be fine because not enough information about the random numbers is revealed to make predictability even come into play. We are indicating this may not always be true: the quality of random bits matters even when the only thing an adversary sees is the result of a one-way functions on these bits.

A common pseudo-random number generator that comes standard with various operating systems is a linear congruential generator with modulus 2^{32}. It is plausible that there are DSA implementations available where the k values are formed by concatenating 5 consecutive outputs from such a generator. Our attack easily extends to this case.

2 Preliminaries

2.1 The Digital Signature Standard

The Digital Signature Standard (DSS, see [16]) is an ElGamal-like [6] digital signature algorithm based on the hardness of computing the discrete logarithm in some finite fields.

THE SCHEME. The scheme uses the following parameters: a prime number p, a prime number q which divides $p-1$ and an element $g \in Z_p^*$ of order q. (Chosen as $g = h^{(p-1)/q}$ where h is a generator of the cyclic group Z_p^*). These parameters may be common to all users of the signature scheme and we will consider them as fixed in the rest of the paper. The standard asks that $2^{159} < q < 2^{160}$ and $p > 2^{511}$. We let $G = \{ g^\alpha : \alpha \in Z_q \}$ denote the subgroup generated by g. Note it has prime order, and that the exponents are from a field, namely Z_q.

The secret key of a user is a random integer x in the range $\{0, \ldots, q-1\}$, and the corresponding public key is $y = g^x \bmod p$. DSA (the Digital Signature Algorithm that underlies the standard) can be used to sign any message $m \in Z_q$, as follows. The signer generates a random number $k \in \{1, \ldots, q-1\}$, which we call the *nonce*. It then computes the values $\lambda = g^k \bmod p$ and $r = \lambda \bmod q$. It sets $s = (xr + m) \cdot k^{-1} \bmod q$, where k^{-1} is the multiplicative inverse of k in the group Z_q^*. The signature of message m is the pair (r, s) and will be denoted by $\mathrm{DSA}(x, k, m)$. Note that a new, random nonce is chosen for each signature.

A purported signature (r, s) of message m can be verified, given the user's public key y, by computing the values $u_1 = m \cdot s^{-1} \bmod q$, $u_2 = r \cdot s^{-1} \bmod q$ and checking that $(g^{u_1} y^{u_2} \bmod p) \bmod q = r$. Notice that the values (r, s) output by $\mathrm{DSA}(x, k, m)$ satisfy the relation $sk - rx = m \pmod q$. We will make use of this relation in our attack on the DSS.

HASHING. The 160-bit "message" m above is not the actual text one wants to sign, but rather the hash of it, under a strong, collision resistant cryptographic hash function H. Specifically, if m is the actual text to be signed, the standard sets H = SHA-1, the Secure Hash Algorithm of [15]. The hashing serves two purposes. The first is to enable one to sign messages of length longer than 160 bits. Second, it "randomizes" the message to prevent any possible attacks based on the algebraic structure of the scheme. Accordingly, following [2], we treat the hash function as a random oracle.

We stress that we are considering *attacks*. In this context, treating H as a random oracle only strengthens our results. If the scheme is breakable when H is a random oracle, we should definitely consider it insecure, because a random oracle is the "best" possible hash function!

Our attack on the DSS algorithm does not involve the hash function H other than to assume it random. Therefore we will assume that the messages are already integers in the range $\{0, \ldots, q-1\}$ and that they are randomly distributed.

SECRECY OF THE NONCE. Recall that for every signature, the signer generates a new, random nonce k. An important feature (drawback!) of the DSS is that the security relies on the secrecy of the nonces. If any nonce k ever becomes revealed, at any time, even long after the signature (r, s) was generated, then given the nonce and the signature one can immediately recover the secret key x, via $x = (sk - m)r^{-1} \bmod q$. This is a key point in our attack.

2.2 Pseudo-Random Number Generators

Each time DSA is used to digitally sign a message m, a nonce k is needed. Ideally k should be a truly random number. In practice the nonces k are pseudo-random numbers produced by a pseudo-random number generator.

A pseudo-random number generator is a program $\mathcal{G}$ that on input a *seed* σ, generates a seemingly random sequence of numbers $\mathcal{G}(\sigma) = k_1, k_2, \ldots$.

The DSS algorithm can be used in conjunction with a pseudo-random number generator as follows. On input a secret key x, a seed σ to the generator, and a sequence of messages $m_1, \ldots, m_n$, run $\mathcal{G}(\sigma)$ to generate a sequence of pseudo-random numbers $k_1, \ldots, k_n$ and run DSA on input x, m_i, k_i for all $i = 1, \ldots, n$.

The pseudo-random number generators we consider in this paper are all variants of the linear congruential generator.

LINEAR CONGRUENTIAL GENERATORS. A linear congruential generator (LCG) is parameterized by a modulus M and two numbers $a, b \in Z_M$. The seed to $\mathcal{G}$ is just a number $\sigma = k_0 \in Z_M$. On input k_0, the generator produces a sequence of numbers, $\mathcal{G}(k_0) = k_1, k_2, \ldots$ defined by the linear recurrence $k_{i+1} = ak_i + b \bmod M$. The values k_i can be directly used by DSA as random nonces to sign the messages. (In which case they are treated modulo q. We assume that with high probability a k_i value will not be 0.)

TRUNCATED LINEAR CONGRUENTIAL GENERATORS. For security reasons it has been suggested that only some of the bits of the number produced by a linear congruential generator be used by applications, in our case the DSA algorithm. A truncated linear congruential generator does exactly this. Let's look at this more closely. A truncated linear congruential generator is parameterized by a modulus M, two numbers $a, b \in Z_M$ and two indices l, h such that $0 \leq l \leq h \leq \lg M$. The seed is a number $\sigma_0 \in Z_M$ and the generator produces numbers in the range $\{0, \ldots, 2^{h-l} - 1\}$. The generator computes a sequence σ_i according to the linear recurrence $\sigma_{i+1} = a\sigma_i + b \bmod M$. Then, each number σ_i is truncated by taking only bits $l, \ldots, h-1$ of the number, to get the number $k_i = ((\sigma_i - (\sigma_i \bmod 2^l)) \bmod 2^h)/2^l$ which is output by the generator.

3 The attack

We look at the security of the DSS when the nonces are generated using a LCG with parameters a, b, M. Later we will extend this to truncated LCGs.

3.1 Overview

Our attack on DSS exploits the relationship $sk - rx = m \bmod q$ holding for any digital signature $(r, s) = \mathrm{DSA}(x, k, m)$ produced by the DSA algorithm. The idea is this. Assume that we receive two messages m_1 and m_2 together with their digital signatures $(r_1, s_1) = \mathrm{DSA}(x, k_1, m_1)$ and $(r_2, s_2) = \mathrm{DSA}(x, k_2, m_2)$. We know that $s_1k_1 - r_1x = m_1 \bmod q$ and $s_2k_2 - r_2x = m_2 \bmod q$. The cryptanalyst knows $m_1, r_1, s_1, m_2, r_2, s_2$. He also knows the public parameters p, q, g of the DSS and the public key $y = g^x$ of the signer. What is hidden from him is the

secret key x of the signer, and also the nonces k_1, k_2 which the signer used to produce the signatures.

At this point, the cryptanalyst is not expected to have any way of determining any of the unknowns short of computing discrete logarithms. However, now suppose we know that a linear congruential generator with parameters a, b, M has been used to produce the nonces. We assume the cryptanalyst knows the parameters a, b, M defining the LCG. (They were chosen at random, but then made public.) What is unknown to the cryptanalyst is the seed k_0 used by the signer to start the LCG. Now, we can combine the two signature equations above with the linear congruential equation $k_2 = ak_1 + b \bmod M$. These three equations together yield a system of three modular equations in three unknowns:

$$\begin{cases} s_1 k_1 - r_1 x = m_1 & (\text{mod } q) \\ s_2 k_2 - r_2 x = m_2 & (\text{mod } q) \\ -ak_1 + k_2 = b & (\text{mod } M) \end{cases} \tag{1}$$

Our approach is to try to solve these equations. Note it is a system of simultaneous modular linear equations in different moduli.

This approach at once raises two questions. One, of course, is how to solve such a system. But the other question may need to be addressed first. Namely, even if we solve it, how do we know the solutions we get are the desired ones? That is, there may be many different solutions, and finding a solution to the system (1) does not necessarily imply that we found the right one. (Meaning the one corresponding to the secret key x.)

This worry arises from a feature of this approach that we should highlight. We are not using all available information. We propose to ignore the fact that $r_i = (g^{k_i} \bmod p) \bmod q$. We will simply try to solve the equations, and see what we get. When we are ignoring what may seem a fundamental relation of the DSS signatures, it is not clear why solving the equations will bring us the right solutions: our system of equations might be under-determined.

We will answer this question in Section 3.2, showing that even disregarding the non-linear relationships $r_1 = (g^{k_1} \bmod p) \bmod q$ and $r_2 = (g^{k_2} \bmod p) \bmod q$, the solution to our equations is uniquely determined in most of the cases. Then we can turn to the problem of solving a system of modular linear equations.

If the moduli are the same, $M = q$, the equations can be easily solved by linear algebra. So, it is insecure to use q as the modulus in the LCG. However, if the modulus M is chosen (randomly and) independently from q, as we assume, one might still imagine that the equation $k_2 = ak_1 + b \bmod M$ does not help in finding the secret key because it is in a different modulus and cannot be easily combined with the other equations. In other words, we are faced with solving a system of simultaneous modular linear equations in different moduli. We address this via lattice reduction techniques in Section 3.3.

In later sections we extend the attack to truncated LCGs and also present a general method for solving systems of simultaneous linear modular equations in different modulii.

3.2 The Uniqueness Lemma

In this section we prove that when DSS is used with a pseudo-random number generator, a few signatures are usually enough for the linear equations $s_i k_i - r_i x = m_i$ to uniquely determine the secret key x, disregarding that it must be also that $r_i = g^{k_i} \bmod p \bmod q$. This answers the first question that we posed in section 3.1 and opens up the possibility of breaking DSS by solving a system of linear equations.

We stress this is true for any generator, not just LCG. The generator might be very strong (eg. cryptographically strong) or very weak, it does not matter. The number of signatures needed depends only on the length of the seed of the generator, growing linearly with this.

The statement we make is a probabilistic one: with high probability the system of equations obtained by using DSS with a linear congruential generator has a unique solution. The probability is taken over the choices of the messages to be signed only. (As discussed in Section 2, these are hashes of the real messages under some "strong" one-way hash function, and so considering them random is natural, especially from an attack point of view.) In other words, no matter how we had chosen x and σ, once they are fixed, if the messages m_i are randomly chosen the secret key x is uniquely determined with high probability.

Before stating the lemma we need some definitions. Fix a secret key $x \in Z_q$ of the DSS. Let $\mathcal{G}$ be some generator (not necessarily LCG) and let M be the total number of seeds that $\mathcal{G}$ can take. So we will think of a seed of $\mathcal{G}$ as being in Z_M. Now fix a seed $\sigma \in Z_M$ of the generator $\mathcal{G}$. Let $\mathcal{G}(\sigma) = k_1, k_2, \ldots, k_n$. Fix a message sequence $m_1, \ldots, m_n \in Z_q$ and let $(r_i, s_i) = \mathrm{DSA}(x, k_i, m_i)$ be the signature of m_i using nonce k_i, for $i = 1, \ldots, n$. Let $x' \in Z_q$ and $\sigma' \in Z_M$, and let $\mathcal{G}(\sigma') = k'_1, \ldots, k'_n$. We say (x', σ') is a *false solution* with respect to $x, \sigma, m_1, \ldots, m_n$ if $x \neq x'$ but $s_i k'_i - r_i x' = m_i \bmod q$ for all $i = 1, \ldots, n$. That is, the secret key is not the right one, but the equations work out anyway.

Lemma 1. *Fix a secret key $x \in Z_q$ of the DSS, and a seed $\sigma \in Z_M$ of the generator $\mathcal{G}$. Now, choose n messages $m_1, \ldots, m_n$ uniformly at random from Z_q. The probability, over the choices of the messages only, that there exists some (x', σ') which is a false solution with respect to $x, \sigma, m_1, \ldots, m_n$ is less than Mq^{1-n}. Moreover, the expected number of such false solutions is also less than Mq^{1-n}.*

Proof. Let $k_1, \ldots, k_n = \mathcal{G}(\sigma)$ be the output of the generator on seed σ. Since σ is fixed, so are $k_1, \ldots, k_n$. We will assume these are all in Z_q^*.

Fix $x' \in Z_q$ and $\sigma' \in Z_M$ such that $x \neq x'$, and let $k'_1, \ldots, k'_n = \mathcal{G}(\sigma')$. For this fixed x', σ', and for a fixed i, we claim that the probability, over the choice of m_i, that $s_i k'_i - r_i x' = m_i$, is at most $1/q$. (Here $(r_i, s_i) = \mathrm{DSA}(x, k_i, m_i)$, so that $r_i = g^{k_i}$ is a fixed quantity, while $s_i = (m_i + x r_i) k_i^{-1}$ is a random variable depending on the choice of m_i.) The reason this claim is not entirely obvious is that indeed s_i depends on m_i.

We first note that $s_i k'_i - r_i x' = m_i$ implies $k_i \neq k'_i$. To see this, note $m_i = s_i k_i - r_i x = m_i k'_i - r_i x'$. If $k_i = k'_i$ we would get $s_i k_i - r_i x = s_i k_i - r_i x'$ which yields $x = x'$ because $r_i \neq 0$. But we assumed $x \neq x'$.

Now, note that if $s_i k_i' - r_i x' = m_i$ then it must be that $(m_i + x r_i) k_i^{-1} k_i' - r_i x' = m_i$, or $m_i(1 - k_i^{-1} k_i') = r_i x k_i^{-1} k_i' - r_i x'$. But $k_i \neq k_i'$ implies $1 - k_i^{-1} k_i' \neq 0$ whence

$$m_i = \frac{r_i(x k_i^{-1} k_i' - x')}{1 - k_i^{-1} k_i'} = \frac{r_i(x k_i' - x' k_i)}{k_i - k_i'}.$$

But the right hand side does not depend on the choice of m_i, because all the quantities there are fixed. (We use here that r_i does not depend on m_i, a property of DSS.) This means there is only one value m_i for which the above equation can be true. So if we pick m_i at random from Z_q, there is only a $1/q$ chance that the above equation can be true.

Now since the messages are chosen independently at random, the probability that $s_i k_i' - r_i x' = m_i$ for all $i = 1, \ldots, n$, is q^{-n}. Recall this is for fixed $x' \in Z_q$ $(x' \neq x)$ and $\sigma' \in Z_M$. The probability that there exists x', σ' which is a false solution is thus, by the union bound, at most $(q-1)M \cdot q^{-n} < Mq^{1-n}$. For the claim about the expected number of false solutions, use linearity of expectation instead of the union bound. ▌

Recall these results are true for any pseudo-random number generator $\mathcal{G}$. That is even if $\mathcal{G}$ is cryptographically strong, with high probability there will be only one secret key x and seed σ such that the equations $r_i x' + s_i k_i' = m_i$ are simultaneously satisfied. Clearly if $\mathcal{G}$ is cryptographically strong it will be hard to recover these x and σ from the signatures (r_i, s_i) and messages m_i only. But for the LCG it can be done.

3.3 Solving the equations

Lemma 1 shows that even if $M \neq q$, if M and q have the same size (i.e., $1/2 < M/q < 2$), the system of equations 1 will usually have only a few solutions. Therefore, if we can solve the system of equations we can also retrieve the secret key.

SOLVING VIA INTEGER PROGRAMMING. We remark that systems of linear equations in different moduli can be rewritten as integer programming problems by introducing a new variable for each equation. Since we have a constant number of equations, they can thus be solved using polynomial time algorithms for integer programming in constant dimensions as given in [14,10]. However these algorithms are relatively complex and slow. Instead, we we want to solve more directly and simply. We now present a simple lattice based algorithm that solves our system using a nearest lattice vector approximation algorithm as a subroutine.

THE NEAREST LATTICE VECTOR PROBLEM. Let $B = \{b_1, \ldots, b_n\}$ be a finite set of vectors in $\mathbf{R}^n$. The lattice generated by B is the set of all integer combinations of the vectors in B and is denoted by $L(B)$. Given B and a vector $x \in R^n$ not in $L(B)$, the nearest lattice vector problem asks for a lattice vectors $w \in L(B)$ such that $\|w - x\| = \min_{v \in L(B)} \|v - x\|$. In [1], Babai gave a simple polynomial time algorithm to find an approximate solution to the nearest lattice vector problem: given the basis B and the target vector x, Babai's algorithm returns a

lattice vector w such that $\|w - x\| \leq c \cdot \min_{v \in L(B)} \|v - x\|$, where $c = 2^{n/2}$ is an approximation factor depending only on the dimension of the lattice.

THE LATTICE. In order to solve the system of equations 1, we set up the following lattice. Let $x' = q/2$, $k_1' = k_2' = M/2$ and define also $\gamma_x = \min\{x', q - x'\}$, $\gamma_{k_1} = \min\{k_1', M - k_1'\}$, $\gamma_{k_2} = \min\{k_2', M - k_2'\}$. Consider the lattice L generated by the columns of the matrix

$$B = \begin{bmatrix} -r_1 & s_1 & 0 & q & 0 & 0 \\ -r_2 & 0 & s_2 & 0 & q & 0 \\ 0 & -a & 1 & 0 & 0 & M \\ \gamma_x^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_{k1}^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_{k2}^{-1} & 0 & 0 & 0 \end{bmatrix}.$$

Notice that multiplying the first three columns of the matrix by x, k_1, k_2 and subtracting the appropriate multiples of the remaining columns to perform modular reduction, we obtain the lattice vector

$$X = (m_1, m_2, m_3, x/\gamma_x, k_1/\gamma_{k1}, k_2/\gamma_{k2})^T$$

from which we can easily recover the secret key x.

Running Babai's nearest lattice vector algorithm on lattice $L(B)$ and target vector $Y = (m_1, m_2, m_3, x'/\gamma_x, k_1'/\gamma_{k1}, k_2'/\gamma_{k2})^T$ we obtain a lattice vector W such that $\|Y - W\| < 8\|Y - X\|$. Now, if $|x - x'| < \gamma_x/14$, $|k_1 - k_1'| < \gamma_{k1}/14$ and $|k_2 - k_2'| < \gamma_{k2}/14$, then $\|Y - W\| < 1$ and since the first three entries of W are integers they must coincide with the corresponding entries in Y and we have $W = (m_1, m_2, m_3, x''/\gamma_x, k_1''/\gamma_{k1}, k_2''/\gamma_{k2})^T$ for some x'', k_1'', k_2'' satisfying the equations 1. Moreover the following two inequalities are satisfied

$$x'' = (x'' - x') + x' \geq -\gamma_x + \gamma_x = 0$$
$$x'' = (x'' - x') + x' < \gamma_x + (q - \gamma_x) = q\,.$$

Inequalities $0 \leq k_1'', k_2'' < M$ can be proved analogously.

If the vector W does not have the desired form, that means that our initial guess (x', k_1', k_2') was not a good enough. If this is the case we simply repeat all the above steps with a different value for x', k_1', k_2'. One can check that if we let x' range in the set $\{q/2 \pm (1 - (1 - 1/8)^j)q/2 \mid j = 0, \ldots, 8 \lg q/2\}$, and k_1, k_2 in the set $\{M/2 \pm (1 - (1 - 1/8)^j)M/2 \mid j = 0, \ldots, 8 \lg M/2\}$, there will be some x', k_1', k_2' such that $|x - x'| < \gamma_x/14$, $|k_1 - k_1'| < \gamma_{k1}/14$ and $|k_2 - k_2'| < \gamma_{k2}/14$.

The number of possible x', k_1', k_2' to start with, to be sure of finding a solution to the system is polynomial in $\lg q$ and $\lg M$, so we can try all of them in polynomial time.

Once we have found a solution x', k_1', k_2' to the equations 1, we can check that we actually found the secret key x by computing $g^{x'} \bmod p$ and comparing it with the public key y. If $g^{x'} \bmod p \neq y$, then $x \neq x'$ and we did not found the solution that we wanted. In this case we can use the method just described to find a solution to the equations 1 in the range $0 \leq x < x'$ or $x' < x < q$. Since by Lemma 1 the total number of x such that system 1 has solution is less then

2 on the average, with high probability we will find the right x after one or two steps.

This completes the description of the attack to DSS when used with linear congruential generators.

4 Solving Simultaneous Modular Equations

The technique described in section 3.3 can be generalized to work on arbitrary systems of linear equations in different moduli. These kind of systems arises in the cryptanalysis of DSS when used with more sophisticated pseudo-random number generators, such as truncated linear congruential generators.

In this section we state the problem of solving a system of linear equations in different moduli in its full generality and give an algorithm to find a solution to such a systems. When the number of equations and variables is fixed, the running time of the algorithm is polynomial in the logarithms of all numbers involved in the description of the equations.

We consider the problem of finding "small" solutions to a system of modular linear equations in different moduli. More precisely, let $U_1, \ldots, U_n$ be positive integers and let V_U be the set of vectors $\{\boldsymbol{x} \in Z^n \mid \forall i. |x_i| < U_i\}$. Let also $A = \{a_{i,j}\}$ be an $m \times n$ integer matrix, and $\boldsymbol{b}$ and $\boldsymbol{M}$ be two vectors in Z^m. We want to find an integer vector $\boldsymbol{x} \in V_U$ such that $A \cdot \boldsymbol{x} = \boldsymbol{b} \pmod{\boldsymbol{M}}$, i.e., $|x_i| < U_i$ for all $i = 1, \ldots, n$ and the following modular equations are simultaneously satisfied

$$\begin{aligned} a_{1,1}x_1 + \ldots + a_{1,n}x_n &= b_1 \pmod{M_1} \\ &\vdots \\ a_{m,1}x_1 + \ldots + a_{m,n}x_n &= b_m \pmod{M_m}. \end{aligned}$$

We first assume that the above system has a solution x and that a good approximation to this solution is known and devise a method to find the exact solution.

Definition 2. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two vectors in V_U. We say that vector $\boldsymbol{y}$ c-approximates $\boldsymbol{x}$ iff for all $i = 1, \ldots, n$ we have $|x_i - y_i| < (U_i - |y_i|)/(c\sqrt{n})$.

Lemma 3. *Let c be a constant greater than $2^{(m+n)/2}$. There exists a polynomial time algorithm that on input $U_1, \ldots, U_n, A, \boldsymbol{b}, \boldsymbol{M}$ as above and a c-approximation $\boldsymbol{y}$ to a solution $\boldsymbol{x} \in V_U$ to $A \cdot \boldsymbol{x} = \boldsymbol{b} \pmod{\boldsymbol{M}}$, finds a (possibly different) solution $\boldsymbol{w} \in V_U$ to $A \cdot \boldsymbol{x} = \boldsymbol{b} \pmod{\boldsymbol{M}}$.*

Proof. Let $\Gamma = \{\gamma_{i,j}\}$ be the $n \times n$ diagonal matrix defined by $\gamma_{i,i} = 1/(U_i - |y_i|)$ and let M be the $m \times m$ diagonal matrix whose diagonal entries are $M_1, \ldots, M_m$. Consider the lattice generated by the columns of the matrix $L = \begin{bmatrix} A & M \\ \Gamma & 0 \end{bmatrix}$ and define the vectors

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{b} \\ \Gamma \cdot \boldsymbol{x} \end{bmatrix} \quad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{b} \\ \Gamma \cdot \boldsymbol{y} \end{bmatrix}$$

Notice that $\boldsymbol{X}$ is a lattice vector and

$$\|\boldsymbol{X}-\boldsymbol{Y}\| = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2\gamma_{i,i}^2} \le \sqrt{\sum_{i=1}^{n}\frac{1}{c^2 n}} = \frac{1}{c}.$$

Running Babai's nearest lattice vector algorithm [1] on lattice L and target vector Y we obtain a lattice vector $\boldsymbol{W}$ such that $\|\boldsymbol{W}-\boldsymbol{Y}\| < c\|\boldsymbol{X}-\boldsymbol{Y}\| \le 1$. Since the first m elements of $\boldsymbol{W}$ and $\boldsymbol{Y}$ are integers, they must be the same. So, the vector $\boldsymbol{W}$ is equal to $\begin{bmatrix} \boldsymbol{b} \\ \Gamma\cdot\boldsymbol{w} \end{bmatrix}$ for some integer vector $\boldsymbol{w}$ satisfying $A\cdot\boldsymbol{w} = \boldsymbol{b} \pmod{\boldsymbol{M}}$. It remains to be proved that $\boldsymbol{w} \in V_U$. Now for all $i = 1,\ldots,n$ we have

$$(w_i-y_i)^2\gamma_{i,i}^2 \le \sum_i (w_i-y_i)^2\gamma_{i,i}^2 = \|\boldsymbol{W}-\boldsymbol{Y}\|^2 < 1,$$

so that $|w_i-y_i| < 1/\gamma_{i,i} = U_i - |y_i|$ and by triangular inequality

$$|w_i| \le |y_i| + |w_i-y_i| < |y_i| + U_i - |y_i| = U_i.$$

This proves $\boldsymbol{w} \in V_U$.

We have shown how to solve a system of modular linear equations, given a good approximation to a solution. We now prove that for any fixed n and m there exists a set of vectors $D \subset V_U$ of size polynomial in the $\lg U_i$ such that for any $x \in V_U$ there exists a vector $y \in D$ such that y is a good approximation of x. This gives a polynomial time algorithm to solve modular linear systems in a fixed number of variables and equations.

Lemma 4. *Let $\delta > (1+c\sqrt{n})/2$ and let D be the set $D_1 \times D_2 \times \cdots \times D_n$ where*

$$D_i = \{\pm(1-(1-1/\delta)^j)U_i \mid j = 0,\ldots,\delta\lg U_i\}.$$

Then for any $x \in V_U$ there exists a vector $y \in D$ such that y is a c-approximation of x.

Proof. Clearly it is sufficient to show that for all i and for all $x \in \{-U_i+1,\ldots,U_i-1\}$, there is some y in D_i such that $|x-y| < (U_i-|y|)/(c\sqrt{n})$. Since the set D_i is symmetric with respect to the origin, we can assume without loss of generality that $x \ge 0$. Now, notice that the sequence $y_j = (1-(1-1/\delta)^j)U_i$ is increasing. Moreover $y_0 = 0$ and $y_{\delta\lg U_i} > U_i - 1$. Therefore there exists a $j \in \{0,\ldots,\delta\lg U_i - 1\}$ such that $y_j \le x < y_{j+1}$. Now, let $x' = y_j + U(1-1/\delta)^j/(c\sqrt{n})$. If $x < x'$ then we have $c\sqrt{n}(x-y_j) < U_i(1-1/\delta)^j = U_i - y_j$ and $|x-y_j| \le (U_i-|y_j|)/(c\sqrt{n})$. Otherwise $x \ge x'$ and we have

$$\begin{aligned} c\sqrt{n}(y_{j+1}-x) &< c\sqrt{n}y_{j+1} - c\sqrt{n}y_j - U_i(1-1/\delta)^j \\ &= U_i(1-1/\delta)^{j+1}(c\sqrt{n}/(\delta-1)-1) \\ &< U_i(1-1/\delta)^{j+1} = U_i - y_{j+1} \end{aligned}$$

and $|x-y_{j+1}| \le (U_i-|y_{j+1}|)/(c\sqrt{n})$.

Theorem 5. *There is an algorithm which on input m modular equations in n variables and n positive integers $U_1,\ldots,U_n$, finds a solution $x_1,\ldots,x_n$ to the*

equations such that $|x_i| < U_i$ for all $i = 1, \ldots, n$ and for any fixed n and m the running time of the algorithm is polynomial in the sizes of the numbers.

In the above theorem the interval in which the variables x_i ranges need not be centered around the origin, as if we want $L_i < x_i < U_i$ we can simply substitute $x_i - (U_i + L_i)/2$ for x_i and obtain an equivalent linear system to be solved in the interval $|x_i| < (U_i - L_i)/2$.

Corollary 6. *There is an algorithm which on input m modular equations in n variables and positive integers $L_1, U_1, \ldots, L_n, U_n$, finds a solution $x_1, \ldots, x_n$ to the equations such that $L_i < x_i < U_i$ for all $i = 1, \ldots, n$ and for any fixed n and m the running time of the algorithm is polynomial in the sizes of the numbers.*

5 Other Pseudo-Random Number Generators

In section 3.1 we presented an attack to DSS that involves the solution of a system of three modular equations in different moduli. The attack easily extends to any pseudo-random number generator expressible by modular linear equations. As an example we consider truncated linear congruential generator and generators where a long nonce is obtained by concatenating shorter random numbers.

5.1 Truncated Linear Congruential Generators

We recall that a truncated linear congruential generator computes a sequence of values σ_i starting from a seed σ_0 according a modular linear recurrence relation, and for each i outputs a number k_i obtained by taking the bits of σ_i between positions l and h. The computation of the σ_i can be easily be expressed by the equation

$$\sigma_i = a\sigma_{i-1} + b \pmod{M} \qquad (0 \leq \sigma_i < M) \tag{2}$$

where a, b and M are the parameters of the generator. Now let's look at how to express the truncation operation. If $l = 0$, then we can simply write

$$k_i = \sigma_i \pmod{2^h} \qquad (0 \leq k_i < 2^h). \tag{3}$$

If $l \neq 0$ we need two equations. First we extract the l-lowest order bits of σ_i via

$$d_i = \sigma_i \pmod{2^l} \qquad (0 \leq d_i < 2^l). \tag{4}$$

Then we use d_i to zero the l-lowest order bits of σ_i and extract the relevant bits of σ_i via

$$2^l k_i = \sigma_i - d_i \pmod{2^h} \qquad (0 \leq k_i < 2^{h-l}). \tag{5}$$

Notice that $\sigma_i - d_i$ is always an integer multiple of 2^l, so equation (5) has solution despite of the fact that 2^l has not an inverse modulo 2^h.

So, the entire process of computing $k_1, k_2, \ldots, k_n$ from a seed σ_0 can be expressed by modular linear equations (2),(4) and (5) for $i = 1, \ldots, n$.

Consider now the use of DSA with a truncated linear congruential generator $\mathcal{G}$ of parameters M, a, b, l, h. For concreteness, we assume that half of the bits are truncated, i.e., $h - l = (\lg M)/2$. Since we want to use the numbers output by

the generator as nonces in the DSA algorithm, we also assume that $h - l = \lg q$. Consider the system of equations

$$\begin{cases} s_1 k_1 - r_1 x = m_1 & (\text{mod } q) \\ s_2 k_2 - r_2 x = m_2 & (\text{mod } q) \\ s_3 k_3 - r_3 x = m_3 & (\text{mod } q) \\ s_4 k_4 - r_4 x = m_4 & (\text{mod } q) \end{cases}$$

together with equations (2),(4) and (5) for $i = 1, \ldots, 4$.

By Corollary 6, we can find in polynomial time a solution to the above equations such that $0 \leq x, k_i < q$, $0 \leq \sigma_i < M$ and $0 \leq d_i < 2^l$. By Lemma 1 we know that with probability $1 - M/q^3 > 1 - 2q^2/q^3 = 1 - 2/q$ the equations have no false solution. Therefore, with high probability, the solution x we found is the DSA secret key.

5.2 Linear Congruential Generators with Concatenation

If the numbers σ_i output by the pseudo-random generator are too short, the nonces k_i can be obtained by concatenating several σ_i together.

For example, a common pseudo-random number generator that comes standard with various operating systems is a linear congruential generator with modulus 2^{32}. The 160 bit number k required to sign with DSA can be obtained by concatenating 5 consecutive outputs from such a generator.

Our attack immediately applies to these schemes. Let σ_i be the sequence of random numbers defined by a linear congruential generator modulo $M = 2^{32}$.

The concatenation operation is easily expressed as a linear equation:

$$k_j = \sigma_{\alpha j} + M\sigma_{\alpha j+1} + \cdots + M^{\alpha-1}\sigma_{\alpha j+\alpha-1} \quad (\text{mod } M^\alpha) \qquad (0 \leq k_i < M^\alpha) \quad (6)$$

where $\alpha = \lceil \lg q / \lg M \rceil = 5$.

This time just two signature equations are enough to guarantee uniqueness of the solution with probability $1 - M/q \approx 1 - 2^{-128}$, and the secret key x can be easily found solving the system of modular equations

$$\begin{cases} s_1 k_1 - r_1 x = m_1 & (\text{mod } q) \\ s_2 k_2 - r_2 x = m_2 & (\text{mod } q) \end{cases}$$

together with equations (2) and (6) for $i = 1, \ldots, 2\alpha - 1$ and $j = 1, 2$.

The attack easily generalizes to generators involving any combination of truncation and concatenation operations.

Acknowledgments

The first author is supported in part by a 1996 Packard Foundation Fellowship in Science and Engineering, and NSF CAREER award CCR-9624439. The second and third authors are supported in part by DARPA contract DABT63-96-C-0018.

References

1. L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

2. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. *Proceedings of the First Annual Conference on Computer and Communications Security*, ACM, 1993.
3. M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Computing*, 13(4):850–863, November 1984.
4. Joan Boyar. Inferring sequences produced by pseudo-random number generators. *Journal of the ACM*, 36(1):129–141, January 1989.
5. A. M. Frieze, R. Kannan, and J. C. Lagarias. Linear congruential generators do not produce random sequences. In *Proc. 25th IEEE Symp. on Foundations of Comp. Science*, pages 480–484, Singer Island, 1984. IEEE.
6. Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. C. Chaum, editors, *Proc. CRYPTO 84*, pages 10–18. Springer, 1985. Lecture Notes in Computer Science No. 196.
7. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. In *Proc. 25th IEEE Symp. on Foundations of Comp. Science*, pages 464–479, Singer Island, 1984. IEEE.
8. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences* 28:270–299, April 1984.
9. J. Hastad and A. Shamir. The cryptographic security of truncated linearly related variables. In *Proc. 17th ACM Symp. on Theory of Computing*, pages 356–362, Providence, 1985. ACM.
10. R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of operations research*, 12(3):415–440, 1987.
11. Donald E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, 1969. Second edition, 1981.
12. Donald E. Knuth. Deciphering a linear congruential encryption. *IEEE Transactions on Information Theory*, IT-31(1):49–52, January 1985.
13. H. Krawczyk. How to predict congruential generators. In G. Brassard, editor, *Proc. CRYPTO 89*, pages 138–153. Springer, 1990. Lecture Notes in Computer Science No. 435.
14. H.W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of operations research*, 8(4):538–548, 1983.
15. National Institute of Standards and Technology (NIST). *FIPS Publication 180: Secure Hash Standard (SHS)*, May 11, 1993.
16. National Institute of Standards and Technology (NIST). *FIPS Publication 186: Digital Signature Standard*, May 19, 1994.
17. J. Plumstead (Boyar). Inferring a sequence generated by a linear congruence. In *Proc. 23rd IEEE Symp. on Foundations of Comp. Science*, pages 153–159, Chicago, 1982. IEEE.
18. Adi Shamir. The generation of cryptographically strong pseudo-random sequences. In Allen Gersho, editor, *Advances in Cryptology: A Report on CRYPTO 81*, pages 1–1. U.C. Santa Barbara Dept. of Elec. and Computer Eng., 1982. Tech Report 82-04.
19. J. Stern. Secret linear congruential generators are not cryptographically secure. In *Proc. 28th IEEE Symp. on Foundations of Comp. Science*, pages 421–426, Los Angeles, 1987. IEEE.
20. A. C. Yao. Theory and application of trapdoor functions. In *Proc. 23rd IEEE Symp. on Foundations of Comp. Science*, pages 80–91, Chicago, 1982. IEEE.

Exhibit 1002

LEDA

A Platform for Combinatorial and Geometric Computing

KURT MEHLHORN
STEFAN NÄHER

CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK http://www.cup.cam.ac.uk
40 West 20th Street, New York, NY 10011-4211, USA http://www.cup.org
10 Stamford Road, Oakleigh, Melbourne 3166, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain

First published 1999

Printed in the United Kingdom at the University Press, Cambridge

Typeset by the author [CRC]

A catalogue record of this book is available from the British Library

Library of Congress cataloguing in publication data applied for

Mehlhorn, Kurt, 1949–
Leda: a platform for combinatorial and geometric computing /
Kurt Mehlhorn, Stefan Näher.
p. cm.
Includes bibliographical references (p.
ISBN 0 521 56329 1 (hb)
1. C++ (Computer program language). 2. LEDA (Computer file)
I. Näher, Stefan. II. Title.
QA76.73.C153M44 2000
005.13'3–dc21 99-24952 CIP

ISBN 0 521 563291 hardback

Disclaimer of Warranty

The programs listed in this book are provided 'as is' without warranty of any kind. We make no warranties, express or implied, that the programs are free of error, or are consistent with any particular standard of merchantability, or that they will meet your requirements for any particular application. They should not be relied upon for solving a problem whose incorrect solution could result in injury or loss of property. If you do use the programs or procedures in such a manner, it is at your own risk. The authors and publisher disclaim all liability for direct, incidental or consequential damages resulting from your use of the programs in this book.

To the best of the authors' and publisher's knowledge, none of the programs included in this book in itself contains any date-sensitive elements that will cause Year 2000-related problems. However, this statement implies no warranty in this matter on the part of the authors or publisher.

Contents