Iliano Cervesato (Ed.)

Advances in Computer Science – ASIAN 2007

Computer and Network Security

12th Asian Computing Science Conference
Doha, Qatar, December 2007
Proceedings

LNCS 4846

Springer

Iliano Cervesato (Ed.)

Advances in Computer Science – ASIAN 2007

Computer and Network Security

12th Asian Computing Science Conference
Doha, Qatar, December 9-11, 2007
Proceedings

Springer

Volume Editor

Iliano Cervesato
Carnegie Mellon University
Doha, Qatar
E-mail: iliano@cmu.edu

Library of Congress Control Number: 2007939450

CR Subject Classification (1998): F.3, E.3, D.2.4, D.4.6-7, K.6.5, C.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-76927-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-76927-9 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12195626 06/3180 5 4 3 2 1 0

Lecture Notes in Computer Science 4846

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Preface

The ASIAN conference series provides a forum for researchers throughout Asia to present cutting-edge results in yearly-themed areas of computer science, to discuss advances in these fields, and to promote interaction with researchers from other continents. Accordingly, the conference moves every year to a different center of research throughout Asia: previous editions were held in Tokyo, Kunming (China), Bangkok, Mumbai, Hanoi, Penang (Malaysia), Phuket (Thailand), Manila, Kathmandu, Singapore, and Pathumthani (Thailand) where ASIAN was initiated by AIT, INRIA and UNU/IIST in 1995. The 12th edition took place in Doha, Qatar, during December 9–11, 2007.

Each year, the conference focuses on a different theme at the cutting edge of computer science research. The theme of ASIAN 2007 was "Computer and Network Security". It has been a tradition of ASIAN to invite three of the most influential researchers in the focus area, one from Asia, one from Europe and one from the Americas, to discuss their work and their vision for the field. This year's distinguished speakers were Andrei Sabelfeld (Chalmers University, Sweden), Joshua Guttman (MITRE, USA) and Kazuhiko Kato (University of Tsukuba, Japan).

Following the call for paper, ASIAN 2007 received 112 submissions, of which 65 were eventually reviewed. Of these, the Program Committee selected 15 regular papers and 10 short papers. This volume contains the abstracts of the invited talks and the revised versions of the regular papers and the short papers. I wish to thank the members of the Program Committee and the external reviewers for doing an excellent job at selecting the contributed papers under severe time pressure. *EasyChair* proved an egregious platform for smoothly carrying out all aspects of the program selection and finalization.

The conference was held in Doha, Qatar, where Carnegie Mellon University recently established a branch campus with the goal of promoting the same high standards of research and education for which its original campus in Pittsburgh, USA, is internationally recognized. Carnegie Mellon Qatar is located in Education City, a 2,500-acre campus which provides state-of-the-art research and teaching facilities to branches of five of the world's leading universities. It is part of an unprecedented commitment of resources made by the Qatari leadership to position Qatar as a world-class center of education and research.

Many people were involved in the organization of this conference. In particular, I wish to thank the General Chair, Kazunori Ueda, for his support, and the Steering Committee for endorsing the candidacy of Doha for this year's edition of ASIAN. This conference would not have been possible without the hard work of the many people who relentlessly handled the local arrangements, especially Thierry Sans and Kara Nesimiuk. We greatly appreciate the generous support

of our sponsors, Carnegie Mellon University in Qatar and QCERT. Finally we are grateful to the authors, the invited speakers and the attendees who made this conference an enjoyable and fruitful event.

September 2007 Iliano Cervesato

Conference Organization

Steering Committee

Philippe Codognet (French Embassy, Japan)
Joxan Jaffar (National University, Singapore)
Mitsu Okada (Keio Univerity, Japan)
R.K. Shyamasundar (Tata Institute of Fundamental Research, India)
Kazunori Ueda (Waseda University, Japan)

General Chair

Kazunori Ueda (Waseda University, Japan)

Program Chair

Iliano Cervesato (Carnegie Mellon University, Qatar)

Program Committee

Michael Backes (Saarland University, Germany)
Anupam Datta (Stanford University, USA)
Mourad Debbabi (Concordia University, Canada)
Sven Dietrich (CERT, USA)
Masami Hagiya (University of Tokyo, Japan)
Yassine Lakhnech (VERIMAG, France)
Ninghui Li (Purdue University, USA)
Catherine Meadows (Naval Research Lab, USA)
R. Ramanujam (Institute of Mathematical Sciences, India)
Takamichi Saito (Meiji University, Japan)
Dheeraj Sanghi (IIT Kanpur, India)
Thierry Sans (Carnegie Mellon University, Qatar)
Andre Scedrov (University of Pennsylvania, USA)
Vitaly Shmatikov (University of Texas-Austin, USA)
Duminda Wijesekera (George Mason University, USA)
Yuqing Zhang (Chinese Academy of Sciences, China)
Jianying Zhou (Institute for Infocomm Research, Singapore)

Local Organization

Thierry Sans (Carnegie Mellon University, Qatar)

External Reviewers

Kumar Avijit
Vishwas B.C.
Adam Barth
A. Baskar
Justin Brickell
Judicaël Courant
Shruti Dubey
Markus Dürmuth
Jason Franklin
Yoshinobu Kawabe
Dilsun Kaynar
Ken Mano
Azzam Mourad
Hadi Otrok
Iosif Radu
Arun Raghavan
Arnab Roy
Hideki Sakurada
Mohamed Saleh
Satyam Sharma
S.P. Suresh
Yasuyuki Tsukada

Lecture Notes in Computer Science

Sublibrary 1: Theoretical Computer Science and General Issues

For information about Vols. 1– 4494
please contact your bookseller or Springer

Vol. 4847: M. Xu, Y. Zhan, J. Cao, Y. Liu (Eds.), Advanced Parallel Processing Technologies. XIX, 767 pages. 2007.

Vol. 4846: I. Cervesato (Ed.), Advances in Computer Science - ASIAN 2007. XI, 313 pages. 2007.

Vol. 4838: T. Masuzawa, S. Tixeuil (Eds.), Stabilization, Safety, and Security of Distributed Systems. XIII, 409 pages. 2007.

Vol. 4783: J. Holub, J. Žďárek (Eds.), Implementation and Application of Automata. XIII, 324 pages. 2007.

Vol. 4782: R. Perrott, B.M. Chapman, J. Subhlok, R.F. de Mello, L.T. Yang (Eds.), High Performance Computing and Communications. XIX, 823 pages. 2007.

Vol. 4771: T. Bartz-Beielstein, M.J. Blesa Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, M. Sampels (Eds.), Hybrid Metaheuristics. X, 202 pages. 2007.

Vol. 4770: V.G. Ganzha, E.W. Mayr, E.V. Vorozhtsov (Eds.), Computer Algebra in Scientific Computing. XIII, 460 pages. 2007.

Vol. 4763: J.-F. Raskin, P.S. Thiagarajan (Eds.), Formal Modeling and Analysis of Timed Systems. X, 369 pages. 2007.

Vol. 4746: A. Bondavalli, F. Brasileiro, S. Rajsbaum (Eds.), Dependable Computing. XV, 239 pages. 2007.

Vol. 4743: P. Thulasiraman, X. He, T.L. Xu, M.K. Denko, R.K. Thulasiram, L.T. Yang (Eds.), Frontiers of High Performance Computing and Networking ISPA 2007 Workshops. XXIX, 536 pages. 2007.

Vol. 4742: I. Stojmenovic, R.K. Thulasiram, L.T. Yang, W. Jia, M. Guo, R.F. de Mello (Eds.), Parallel and Distributed Processing and Applications. XX, 995 pages. 2007.

Vol. 4739: R.M. Díaz, F. Pichler, A.Q. Arencibia (Eds.), Computer Aided Systems Theory – EUROCAST 2007. XIX, 1233 pages. 2007.

Vol. 4736: S. Winter, M. Duckham, L. Kulik, B. Kuipers (Eds.), Spatial Information Theory. XV, 455 pages. 2007.

Vol. 4732: K. Schneider, J. Brandt (Eds.), Theorem Proving in Higher Order Logics. IX, 401 pages. 2007.

Vol. 4731: A. Pelc (Ed.), Distributed Computing. XVI, 510 pages. 2007.

Vol. 4726: N. Ziviani, R. Baeza-Yates (Eds.), String Processing and Information Retrieval. XII, 311 pages. 2007.

Vol. 4711: C.B. Jones, Z. Liu, J. Woodcock (Eds.), Theoretical Aspects of Computing – ICTAC 2007. XI, 483 pages. 2007.

Vol. 4710: C.W. George, Z. Liu, J. Woodcock (Eds.), Domain Modeling and the Duration Calculus. XI, 237 pages. 2007.

Vol. 4708: L. Kučera, A. Kučera (Eds.), Mathematical Foundations of Computer Science 2007. XVIII, 764 pages. 2007.

Vol. 4707: O. Gervasi, M.L. Gavrilova (Eds.), Computational Science and Its Applications – ICCSA 2007, Part III. XXIV, 1205 pages. 2007.

Vol. 4706: O. Gervasi, M.L. Gavrilova (Eds.), Computational Science and Its Applications – ICCSA 2007, Part II. XXIII, 1129 pages. 2007.

Vol. 4705: O. Gervasi, M.L. Gavrilova (Eds.), Computational Science and Its Applications – ICCSA 2007, Part I. XLIV, 1169 pages. 2007.

Vol. 4703: L. Caires, V.T. Vasconcelos (Eds.), CONCUR 2007 – Concurrency Theory. XIII, 507 pages. 2007.

Vol. 4700: C.B. Jones, Z. Liu, J. Woodcock (Eds.), Formal Methods and Hybrid Real-Time Systems. XVI, 539 pages. 2007.

Vol. 4699: B. Kågström, E. Elmroth, J. Dongarra, J. Waśniewski (Eds.), Applied Parallel Computing. XXIX, 1192 pages. 2007.

Vol. 4698: L. Arge, M. Hoffmann, E. Welzl (Eds.), Algorithms – ESA 2007. XV, 769 pages. 2007.

Vol. 4697: L. Choi, Y. Paek, S. Cho (Eds.), Advances in Computer Systems Architecture. XIII, 400 pages. 2007.

Vol. 4688: K. Li, M. Fei, G.W. Irwin, S. Ma (Eds.), Bio-Inspired Computational Intelligence and Applications. XIX, 805 pages. 2007.

Vol. 4684: L. Kang, Y. Liu, S. Zeng (Eds.), Evolvable Systems: From Biology to Hardware. XIV, 446 pages. 2007.

Vol. 4683: L. Kang, Y. Liu, S. Zeng (Eds.), Advances in Computation and Intelligence. XVII, 663 pages. 2007.

Vol. 4681: D.-S. Huang, L. Heutte, M. Loog (Eds.), Advanced Intelligent Computing Theories and Applications. XXVI, 1379 pages. 2007.

Vol. 4672: K. Li, C. Jesshope, H. Jin, J.-L. Gaudiot (Eds.), Network and Parallel Computing. XVIII, 558 pages. 2007.

Vol. 4671: V.E. Malyshkin (Ed.), Parallel Computing Technologies. XIV, 635 pages. 2007.

Vol. 4669: J.M. de Sá, L.A. Alexandre, W. Duch, D. Mandic (Eds.), Artificial Neural Networks – ICANN 2007, Part II. XXXI, 990 pages. 2007.

Vol. 4668: J.M. de Sá, L.A. Alexandre, W. Duch, D. Mandic (Eds.), Artificial Neural Networks – ICANN 2007, Part I. XXXI, 978 pages. 2007.

Vol. 4666: M.E. Davies, C.J. James, S.A. Abdallah, M.D. Plumbley (Eds.), Independent Component Analysis and Blind Signal Separation. XIX, 847 pages. 2007.

Vol. 4665: J. Hromkovič, R. Královič, M. Nunkesser, P. Widmayer (Eds.), Stochastic Algorithms: Foundations and Applications. X, 167 pages. 2007.

Vol. 4664: J. Durand-Lose, M. Margenstern (Eds.), Machines, Computations, and Universality. X, 325 pages. 2007.

Vol. 4661: U. Montanari, D. Sannella, R. Bruni (Eds.), Trustworthy Global Computing. X, 339 pages. 2007.

Vol. 4649: V. Diekert, M.V. Volkov, A. Voronkov (Eds.), Computer Science – Theory and Applications. XIII, 420 pages. 2007.

Vol. 4647: R. Martin, M.A. Sabin, J.R. Winkler (Eds.), Mathematics of Surfaces XII. IX, 509 pages. 2007.

Vol. 4646: J. Duparc, T.A. Henzinger (Eds.), Computer Science Logic. XIV, 600 pages. 2007.

Vol. 4644: N. Azémard, L. Svensson (Eds.), Integrated Circuit and System Design. XIV, 583 pages. 2007.

Vol. 4641: A.-M. Kermarrec, L. Bougé, T. Priol (Eds.), Euro-Par 2007 Parallel Processing. XXVII, 974 pages. 2007.

Vol. 4639: E. Csuhaj-Varjú, Z. Ésik (Eds.), Fundamentals of Computation Theory. XIV, 508 pages. 2007.

Vol. 4638: T. Stützle, M. Birattari, H. H. Hoos (Eds.), Engineering Stochastic Local Search Algorithms. X, 223 pages. 2007.

Vol. 4630: H.J. van den Herik, P. Ciancarini, H.H.L.M.(J.) Donkers (Eds.), Computers and Games. XII, 283 pages. 2007.

Vol. 4628: L.N. de Castro, F.J. Von Zuben, H. Knidel (Eds.), Artificial Immune Systems. XII, 438 pages. 2007.

Vol. 4627: M. Charikar, K. Jansen, O. Reingold, J.D.P. Rolim (Eds.), Approximation, Randomization, and Combinatorial Optimization. XII, 626 pages. 2007.

Vol. 4624: T. Mossakowski, U. Montanari, M. Haveraaen (Eds.), Algebra and Coalgebra in Computer Science. XI, 463 pages. 2007.

Vol. 4623: M. Collard (Ed.), Ontologies-Based Databases and Information Systems. X, 153 pages. 2007.

Vol. 4621: D. Wagner, R. Wattenhofer (Eds.), Algorithms for Sensor and Ad Hoc Networks. XIII, 415 pages. 2007.

Vol. 4619: F. Dehne, J.-R. Sack, N. Zeh (Eds.), Algorithms and Data Structures. XVI, 662 pages. 2007.

Vol. 4618: S.G. Akl, C.S. Calude, M.J. Dinneen, G. Rozenberg, H.T. Wareham (Eds.), Unconventional Computation. X, 243 pages. 2007.

Vol. 4616: A.W.M. Dress, Y. Xu, B. Zhu (Eds.), Combinatorial Optimization and Applications. XI, 390 pages. 2007.

Vol. 4614: B. Chen, M. Paterson, G. Zhang (Eds.), Combinatorics, Algorithms, Probabilistic and Experimental Methodologies. XII, 530 pages. 2007.

Vol. 4613: F.P. Preparata, Q. Fang (Eds.), Frontiers in Algorithmics. XI, 348 pages. 2007.

Vol. 4600: H. Comon-Lundh, C. Kirchner, H. Kirchner (Eds.), Rewriting, Computation and Proof. XVI, 273 pages. 2007.

Vol. 4599: S. Vassiliadis, M. Bereković, T.D. Hämäläinen (Eds.), Embedded Computer Systems: Architectures, Modeling, and Simulation. XVIII, 466 pages. 2007.

Vol. 4598: G. Lin (Ed.), Computing and Combinatorics. XII, 570 pages. 2007.

Vol. 4596: L. Arge, C. Cachin, T. Jurdziński, A. Tarlecki (Eds.), Automata, Languages and Programming. XVII, 953 pages. 2007.

Vol. 4595: D. Bošnački, S. Edelkamp (Eds.), Model Checking Software. X, 285 pages. 2007.

Vol. 4590: W. Damm, H. Hermanns (Eds.), Computer Aided Verification. XV, 562 pages. 2007.

Vol. 4588: T. Harju, J. Karhumäki, A. Lepistö (Eds.), Developments in Language Theory. XI, 423 pages. 2007.

Vol. 4583: S.R. Della Rocca (Ed.), Typed Lambda Calculi and Applications. X, 397 pages. 2007.

Vol. 4580: B. Ma, K. Zhang (Eds.), Combinatorial Pattern Matching. XII, 366 pages. 2007.

Vol. 4576: D. Leivant, R. de Queiroz (Eds.), Logic, Language, Information and Computation. X, 363 pages. 2007.

Vol. 4547: C. Carlet, B. Sunar (Eds.), Arithmetic of Finite Fields. XI, 355 pages. 2007.

Vol. 4546: J. Kleijn, A. Yakovlev (Eds.), Petri Nets and Other Models of Concurrency – ICATPN 2007. XI, 515 pages. 2007.

Vol. 4545: H. Anai, K. Horimoto, T. Kutsia (Eds.), Algebraic Biology. XIII, 379 pages. 2007.

Vol. 4533: F. Baader (Ed.), Term Rewriting and Applications. XII, 419 pages. 2007.

Vol. 4528: J. Mira, J.R. Álvarez (Eds.), Nature Inspired Problem-Solving Methods in Knowledge Engineering, Part II. XXII, 650 pages. 2007.

Vol. 4527: J. Mira, J.R. Álvarez (Eds.), Bio-inspired Modeling of Cognitive Tasks, Part I. XXII, 630 pages. 2007.

Vol. 4525: C. Demetrescu (Ed.), Experimental Algorithms. XIII, 448 pages. 2007.

Vol. 4514: S.N. Artemov, A. Nerode (Eds.), Logical Foundations of Computer Science. XI, 513 pages. 2007.

Vol. 4513: M. Fischetti, D.P. Williamson (Eds.), Integer Programming and Combinatorial Optimization. IX, 500 pages. 2007.

Vol. 4510: P. Van Hentenryck, L.A. Wolsey (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. X, 391 pages. 2007.

Vol. 4507: F. Sandoval, A.G. Prieto, J. Cabestany, M. Graña (Eds.), Computational and Ambient Intelligence. XXVI, 1167 pages. 2007.

Vol. 4502: T. Altenkirch, C. McBride (Eds.), Types for Proofs and Programs. VIII, 269 pages. 2007.

Vol. 4501: J. Marques-Silva, K.A. Sakallah (Eds.), Theory and Applications of Satisfiability Testing – SAT 2007. XI, 384 pages. 2007.

Vol. 4497: S.B. Cooper, B. Löwe, A. Sorbi (Eds.), Computation and Logic in the Real World. XVIII, 826 pages. 2007.

Table of Contents

Invited Speaker: Joshua Guttman

Session 4: Protocols

Invited Speaker: Kazuhiko Kato

Session 5: Intrusion Detection

Session 6: Short Papers on Network Security

Session 7: Safe Execution

Dimensions of Declassification in Theory and Practice

Andrei Sabelfeld

Dept. of Computer Science and Engineering, Chalmers University of Technology
412 96 Gothenburg, Sweden

Abstract. Computing systems often deliberately release (or declassify) sensitive information. A principal security concern for systems permitting information release is whether this release is safe: is it possible that the attacker compromises the information release mechanism and extracts more secret information than intended? While the security community has recognized the importance of the problem, the state-of-the-art in information release is, unfortunately, a number of approaches with somewhat unconnected semantic goals. We provide a road map of the main directions of current research, by classifying the basic goals according to *what* information is released, *who* releases information, *where* in the system information is released, and *when* information can be released. We apply this classification in order to evaluate the security of a case study realized in a security-typed language: an implementation of a non-trivial cryptographic protocol that allows playing online poker without a trusted third party. In addition, we identify some prudent principles of declassification. These principles shed light on existing definitions and may also serve as useful "sanity checks" for emerging models.

The talk is based on joint work, in part, with David Sands, and, in part, with Aslan Askarov.

I. Cervesato (Ed.): ASIAN 2007, LNCS 4846, p. 1, 2007.

A Static Birthmark of Binary Executables Based on API Call Structure*

Seokwoo Choi, Heewan Park, Hyun-il Lim, and Taisook Han

Division of Computer Science and
Advanced Information Technology Research Center(AITrc).
Korea Advanced Institute of Science and Technology
{swchoi,hwpark,hilim}@pllab.kaist.ac.kr, han@cs.kaist.ac.kr

Abstract. A software birthmark is a unique characteristic of a program that can be used as a software theft detection. In this paper we suggest and empirically evaluate a static birthmark of binary executables based on API call structure. The program properties employed in this birthmark are functions and standard API calls when the functions are executed. The API calls from a function includes the API calls explicitly found from the function and its descendants within limited depth in the call graph. To statically identify functions, call graphs and API calls, we utilizes IDAPro disassembler and its plug-ins. We define the similarity between two functions as the proportion of the number of all API calls to the number of the common API calls. The similarity between two programs is obtained by the maximum weight bipartite matching between two programs using the function similarity matrix. To show the credibility of the proposed techniques, we compare the same applications with different versions and the various types of applications which include text editors, picture viewers, multimedia players, P2P applications and ftp clients. To show the resilience, we compare binary executables compiled from various compilers. The empirical result shows that the similarities obtained using our birthmark sufficiently indicates the functional and structural similarities among programs.

Keyword: software piracy, software birthmark, binary analysis.

1 Introduction

Recently a large amount of software is developed in the form of open source projects. Most open source projects contain software licenses. A widely used software license for open source software is the GNU Public License(GPL). The GPL allows developers to use software freely, but requires new projects using the original work to be licensed under the GPL. There are also more permissive software licenses like the MIT license and the BSD licenses which allow the original source code to be combined in commercial software. The permissive licenses, however, require the copyright notice of the original software to be included.

* This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center(AITrc).

I. Cervesato (Ed.): ASIAN 2007, LNCS 4846, pp. 2–16, 2007.

There have been reported that many companies use open source software for commercial purpose without permission. To detect code theft when source code is available, we can utilize well-known plagiarism detection tools like MOSS, JPlag and YAP [1,2,3]. Suppose that source code under the GPL is contained in commercial software, which is distributed in compiled binaries without indicating the copyright notice of the original software. In this case, we need to prove whether the open source code is used or not in the binary executables. Software birthmarking is one of the techniques to solve such software theft problems.

A software birthmark is unique characteristics of a program that can be used to identify the program. If program p and q have the same or very similar birthmark, q is very likely to be a stolen copy of p (and vice versa). Comparing the strings analyzed from binary executables can be a easy birthmarking technique. In this case, a set of strings is a birthmark. Sometimes comparing the structures of binaries can be a good birthmark technique. For example, SabreSecurity BinDiff effectively found similarities between two MacOS emulators named CherryOS and PearPC[4,5]. Tamada et al. suggested a dynamic software birthmark for Windows applications using Win32 API function call sequences [6,7]. Dynamic birthmarks extract program properties from a program execution trace when a sequence of input is given, while static birthmarks extract properties only from the program itself.

In this paper we propose a new static birthmarking technique that can help to identify ownership and similarity of binary executables. Program properties used as our birthmark are summaries extracted from each binary function in a program. The summary of each function is a set of possible standard API calls when the function is executed. We statically identify API function calls by analyzing disassembled code which is generated by IDAPro disassembler[8]. A similarity between two functions is calculated by comparing API call sets of two functions. A similarity between two programs is obtained by matching problem.

We evaluate the proposed birthmark by comparing various categories of Windows applications. To show the credibility, the same applications with different versions are compared. To show the resilience, we compare binary executables compiled from various compilers. The empirical result shows that the similarities obtained using our birthmark sufficiently indicate the functional and structural similarities among programs.

2 Related Work

There are three major threats against the intellectual property contained in software. *Software piracy* is the illegal use, duplication or reselling of legally protected software. *Software tampering* is the illegal modification of software to gain control over restricted code or digital media protected by the software. *Malicious reverse engineering* is the extracting of a piece of a program in order to reuse it in one's own. To deal with these threats, several techniques have been explored, for example, software watermarking to deal with piracy, code obfuscation to deter reverse engineering, and software tamper-proofing [9].

Software watermarking is a well-known technique used to provide a way to prove ownership of stolen software. Software watermarking systems embed watermarks in software and recognize the watermarks. Software watermark can be either static or dynamic [10,11]. Unfortunately, watermarking is not always feasible because it requires software developers to embed a watermark before releasing the software.

Software birthmarking is a technique that identifies the inherent characteristics occurring in a program by chance. Unlike software watermarks, software birthmarks do not embed additional code or identifier. Instead a birthmark relies on an inherent characteristic of the application to show that one program is a copy of another. The result of comparing two programs with software birthmarking is similarities between two programs. With the similarities, we are able to say that one program is a copy of another, totally or in part.

Tamada et al. [12,13] suggested the first practical application of static software birthmarks to identify the theft of programs. This technique is specific to Java class files which is a combination of four individual birthmarks: constant values in field variables(CVFV), sequence of method calls(SMC), inheritance structure(IS), and used classes(UC). These four birthmarks could be used individually but the combination makes this technique more reliable. Their experiment with several sample programs shows that the proposed birthmarks identify a class within a program with high precision, but can easily be confused by several obfuscation techniques.

Tamada et al. [6,7] introduced dynamic birthmarks and proposed two birthmarks based on the trace of system calls for Windows programs. The dynamic birthmarks are the sequence and frequency of API function calls during execution of software. They claim that these birthmarks are reasonably robust against program transformations. The credibility of this birthmark highly relies on user interactions, inputs and system environments. To avoid this weakness, they highly restricted inputs and user interactions in the experiments.

Myles et al. proposed a k-gram based static birthmark [14]. They adopted k-gram, which have been previously used to detect similarity between documents, as their birthmark for Java applications. The k-gram birthmark is the set of unique opcode sequence of length k. For each method in a module they compute the set of unique k-grams by sliding a window of length k over the static instruction sequence. k-gram based birthmark is precise, but highly susceptible to program transformations. They evaluated this birthmarking techniques with several tiny Java programs.

Myles et al. [15,16] proposed the concept of another dynamic birthmark known as Whole Program Path(WPP) birthmark. A WPP is a directed acyclic graph(DAG) representation of a context-free grammar that generates a program's acyclic path [17]. To get WPP, dynamic trace of a program is obtained by instrumentation, and the trace is compressed into a DAG using SEQUITUR algorithm. They used WPP as their birthmarks and computed similarity between two birthmarks using a graph distance for maximal common subgraph [18]. They experimented WPP birthmarking technique with a few tiny Java programs. The