Lecture Notes in
Computer Science 773

Douglas R. Stinson (Ed.)

Advances in Cryptology – CRYPTO '93

13th Annual International Cryptology Conference
Santa Barbara, California, USA, August 1993
Proceedings

Springer-Verlag

Douglas R. Stinson (Ed.)

Advances in Cryptology – CRYPTO '93

13th Annual International Cryptology Conference
Santa Barbara, California, USA
August 22-26, 1993
Proceedings

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Volume Editor

Douglas R. Stinson
Computer Science and Engineering Department
and Center for Communication and Information Science
University of Nebraska, Lincoln, NE 68588-01115, USA

CR Subject Classification (1991): E.3-4, D.4.6, G.2.1

ISBN 3-540-57766-1 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-57766-1 Springer-Verlag New York Berlin Heidelberg

Printed in Germany

Typesetting: Camera-ready by author
SPIN: 10131900 45/3140-543210 - Printed on acid-free paper

Lecture Notes in Computer Science 773

Edited by G. Goos and J. Hartmanis

PREFACE

The CRYPTO '93 conference was sponsored by the International Association for Cryptologic Research (IACR) and Bell-Northern Research (a subsidiary of Northern Telecom), in co-operation with the IEEE Computer Society Technical Committee. It took place at the University of California, Santa Barbara, from August 22–26, 1993. This was the thirteenth annual CRYPTO conference, all of which have been held at UCSB. The conference was very enjoyable and ran very smoothly, largely due to the efforts of the General Chair, Paul Van Oorschot. It was a pleasure working with Paul throughout the months leading up to the conference.

There were 136 submitted papers which were considered by the Program Committee. Of these, 38 were selected for presentation at the conference. There was also one invited talk at the conference, presented by Miles Smid, the title of which was "A Status Report On the Federal Government Key Escrow System."

The conference also included the customary Rump Session, which was presided over by Whit Diffie in his usual inimitable fashion. Thanks again to Whit for organizing and running the Rump session. This year, the Rump Session included an interesting and lively panel discussion on issues pertaining to key escrowing. Those taking part were W. Diffie, J. Gilmore, S. Goldwasser, M. Hellman, A. Herzberg, S. Micali, R. Rueppel, G. Simmons and D. Weitzner.

These proceedings contain revised versions of the 38 contributed talks, as well as two talks from the Rump Session. Please remember that these papers are unrefereed, and many of them represent work in progress. Some authors will write final versions of their papers for publication in refereed journals at a later time. Of course, the authors bear full responsibility for the contents of their papers.

I am very grateful to the members of the Program Committee for their hard work and dedication in the difficult task of selecting less than 30% of the submitted papers for presentation at the conference. The members of the program committee were as follows:

Mihir Bellare (IBM T. J. Watson)
Eli Biham (Technion, Israel)
Ernie Brickell (Sandia Laboratories)
Joan Feigenbaum (AT&T Bell Laboratories)
Russell Impagliazzo (UCSD)
Andrew Odlyzko (AT&T Bell Laboratories)
Tatsuaki Okamoto (NTT, Japan)
Birgit Pfitzmann (Hildesheim, Germany)
Rainer Rueppel (R^3, Switzerland)
Scott Vanstone (Waterloo, Canada)

As has been done since 1989, submissions to CRYPTO '93 were required to be anonymous. As well, we followed recent tradition which dictates that Program

Committee members could be an author or co-author of at most one accepted paper. Papers submitted by members of the Program Committee underwent the normal reviewing process (and, of course, no Program Committee member reviewed his or her own paper).

Thanks to Jimmy Upton for help with the pre-proceedings that were distributed at the conference (incidentally, this is the last year that CRYPTO will have both pre-proceedings and proceedings — starting in 1994, the proceedings will be available at the conference). Thanks also to Gus Simmons and Carol Patterson, who helped out with registration at the conference. And I would also like to convey my gratitude to Deb Heckens and my student, K. Gopalakrishnan, for their assistance.

Finally, I would like to thank everyone who submitted talks for CRYPTO '93. It goes without saying that the success of the conference depends ultimately on the quality of the submissions — CRYPTO has been and remains a leading conference in the discipline due the the high quality of the papers. I am also grateful to the authors for sending me final versions of their papers for publication in these proceedings in a timely fashion.

Douglas Stinson
Program Chair, CRYPTO '93
University of Nebraska
November, 1993

Lecture Notes in Computer Science

For information about Vols. 1–693
please contact your bookseller or Springer-Verlag

Vol. 694: A. Bode, M. Reeve, G. Wolf (Eds.), PARLE '93. Parallel Architectures and Languages Europe. Proceedings, 1993. XVII, 770 pages. 1993.

Vol. 695: E. P. Klement, W. Slany (Eds.), Fuzzy Logic in Artificial Intelligence. Proceedings, 1993. VIII, 192 pages. 1993. (Subseries LNAI).

Vol. 696: M. Worboys, A. F. Grundy (Eds.), Advances in Databases. Proceedings, 1993. X, 276 pages. 1993.

Vol. 697: C. Courcoubetis (Ed.), Computer Aided Verification. Proceedings, 1993. IX, 504 pages. 1993.

Vol. 698: A. Voronkov (Ed.), Logic Programming and Automated Reasoning. Proceedings, 1993. XIII, 386 pages. 1993. (Subseries LNAI).

Vol. 699: G. W. Mineau, B. Moulin, J. F. Sowa (Eds.), Conceptual Graphs for Knowledge Representation. Proceedings, 1993. IX, 451 pages. 1993. (Subseries LNAI).

Vol. 700: A. Lingas, R. Karlsson, S. Carlsson (Eds.), Automata, Languages and Programming. Proceedings, 1993. XII, 697 pages. 1993.

Vol. 701: P. Atzeni (Ed.), LOGIDATA+: Deductive Databases with Complex Objects. VIII, 273 pages. 1993.

Vol. 702: E. Börger, G. Jäger, H. Kleine Büning, S. Martini, M. M. Richter (Eds.), Computer Science Logic. Proceedings, 1992. VIII, 439 pages. 1993.

Vol. 703: M. de Berg, Ray Shooting, Depth Orders and Hidden Surface Removal. X, 201 pages. 1993.

Vol. 704: F. N. Paulisch, The Design of an Extendible Graph Editor. XV, 184 pages. 1993.

Vol. 705: H. Grünbacher, R. W. Hartenstein (Eds.), Field-Programmable Gate Arrays. Proceedings, 1992. VIII, 218 pages. 1993.

Vol. 706: H. D. Rombach, V. R. Basili, R. W. Selby (Eds.), Experimental Software Engineering Issues. Proceedings, 1992. XVIII, 261 pages. 1993.

Vol. 707: O. M. Nierstrasz (Ed.), ECOOP '93 – Object-Oriented Programming. Proceedings, 1993. XI, 531 pages. 1993.

Vol. 708: C. Laugier (Ed.), Geometric Reasoning for Perception and Action. Proceedings, 1991. VIII, 281 pages. 1993.

Vol. 709: F. Dehne, J.-R. Sack, N. Santoro, S. Whitesides (Eds.), Algorithms and Data Structures. Proceedings, 1993. XII, 634 pages. 1993.

Vol. 710: Z. Ésik (Ed.), Fundamentals of Computation Theory. Proceedings, 1993. IX, 471 pages. 1993.

Vol. 711: A. M. Borzyszkowski, S. Sokołowski (Eds.), Mathematical Foundations of Computer Science 1993. Proceedings, 1993. XIII, 782 pages. 1993.

Vol. 712: P. V. Rangan (Ed.), Network and Operating System Support for Digital Audio and Video. Proceedings, 1992. X, 416 pages. 1993.

Vol. 713: G. Gottlob, A. Leitsch, D. Mundici (Eds.), Computational Logic and Proof Theory. Proceedings, 1993. XI, 348 pages. 1993.

Vol. 714: M. Bruynooghe, J. Penjam (Eds.), Programming Language Implementation and Logic Programming. Proceedings, 1993. XI, 421 pages. 1993.

Vol. 715: E. Best (Ed.), CONCUR'93. Proceedings, 1993. IX, 541 pages. 1993.

Vol. 716: A. U. Frank, I. Campari (Eds.), Spatial Information Theory. Proceedings, 1993. XI, 478 pages. 1993.

Vol. 717: I. Sommerville, M. Paul (Eds.), Software Engineering – ESEC '93. Proceedings, 1993. XII, 516 pages. 1993.

Vol. 718: J. Seberry, Y. Zheng (Eds.), Advances in Cryptology – AUSCRYPT '92. Proceedings, 1992. XIII, 543 pages. 1993.

Vol. 719: D. Chetverikov, W.G. Kropatsch (Eds.), Computer Analysis of Images and Patterns. Proceedings, 1993. XVI, 857 pages. 1993.

Vol. 720: V.Mařík, J. Lažanský, R.R. Wagner (Eds.), Database and Expert Systems Applications. Proceedings, 1993. XV, 768 pages. 1993.

Vol. 721: J. Fitch (Ed.), Design and Implementation of Symbolic Computation Systems. Proceedings, 1992. VIII, 215 pages. 1993.

Vol. 722: A. Miola (Ed.), Design and Implementation of Symbolic Computation Systems. Proceedings, 1993. XII, 384 pages. 1993.

Vol. 723: N. Aussenac, G. Boy, B. Gaines, M. Linster, J.-G. Ganascia, Y. Kodratoff (Eds.), Knowledge Acquisition for Knowledge-Based Systems. Proceedings, 1993. XIII, 446 pages. 1993. (Subseries LNAI).

Vol. 724: P. Cousot, M. Falaschi, G. Filè, A. Rauzy (Eds.), Static Analysis. Proceedings, 1993. IX, 283 pages. 1993.

Vol. 725: A. Schiper (Ed.), Distributed Algorithms. Proceedings, 1993. VIII, 325 pages. 1993.

Vol. 726: T. Lengauer (Ed.), Algorithms – ESA '93. Proceedings, 1993. IX, 419 pages. 1993

Vol. 727: M. Filgueiras, L. Damas (Eds.), Progress in Artificial Intelligence. Proceedings, 1993. X, 362 pages. 1993. (Subseries LNAI).

Vol. 728: P. Torasso (Ed.), Advances in Artificial Intelligence. Proceedings, 1993. XI, 336 pages. 1993. (Subseries LNAI).

Vol. 729: L. Donatiello, R. Nelson (Eds.), Performance Evaluation of Computer and Communication Systems. Proceedings, 1993. VIII, 675 pages. 1993.

Vol. 730: D. B. Lomet (Ed.), Foundations of Data Organization and Algorithms. Proceedings, 1993. XII, 412 pages. 1993.

Vol. 731: A. Schill (Ed.), DCE – The OSF Distributed Computing Environment. Proceedings, 1993. VIII, 285 pages. 1993.

Vol. 732: A. Bode, M. Dal Cin (Eds.), Parallel Computer Architectures. IX, 311 pages. 1993.

Vol. 733: Th. Grechenig, M. Tscheligi (Eds.), Human Computer Interaction. Proceedings, 1993. XIV, 450 pages. 1993.

Vol. 734: J. Volkert (Ed.), Parallel Computation. Proceedings, 1993. VIII, 248 pages. 1993.

Vol. 735: D. Bjørner, M. Broy, I. V. Pottosin (Eds.), Formal Methods in Programming and Their Applications. Proceedings, 1993. IX, 434 pages. 1993.

Vol. 736: R. L. Grossman, A. Nerode, A. P. Ravn, H. Rischel (Eds.), Hybrid Systems. VIII, 474 pages. 1993.

Vol. 737: J. Calmet, J. A. Campbell (Eds.), Artificial Intelligence and Symbolic Mathematical Computing. Proceedings, 1992. VIII, 305 pages. 1993.

Vol. 738: M. Weber, M. Simons, Ch. Lafontaine, The Generic Development Language Deva. XI, 246 pages. 1993.

Vol. 739: H. Imai, R. L. Rivest, T. Matsumoto (Eds.), Advances in Cryptology – ASIACRYPT '91. X, 499 pages. 1993.

Vol. 740: E. F. Brickell (Ed.), Advances in Cryptology – CRYPTO '92. Proceedings, 1992. X, 593 pages. 1993.

Vol. 741: B. Preneel, R. Govaerts, J. Vandewalle (Eds.), Computer Security and Industrial Cryptography. Proceedings, 1991. VIII, 275 pages. 1993.

Vol. 742: S. Nishio, A. Yonezawa (Eds.), Object Technologies for Advanced Software. Proceedings, 1993. X, 543 pages. 1993.

Vol. 743: S. Doshita, K. Furukawa, K. P. Jantke, T. Nishida (Eds.), Algorithmic Learning Theory. Proceedings, 1992. X, 260 pages. 1993. (Subseries LNAI)

Vol. 744: K. P. Jantke, T. Yokomori, S. Kobayashi, E. Tomita (Eds.), Algorithmic Learning Theory. Proceedings, 1993. XI, 423 pages. 1993. (Subseries LNAI)

Vol. 745: V. Roberto (Ed.), Intelligent Perceptual Systems. VIII, 378 pages. 1993. (Subseries LNAI)

Vol. 746: A. S. Tanguiane, Artificial Perception and Music Recognition. XV, 210 pages. 1993. (Subseries LNAI).

Vol. 747: M. Clarke, R. Kruse, S. Moral (Eds.), Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Proceedings, 1993. X, 390 pages. 1993.

Vol. 748: R. H. Halstead Jr., T. Ito (Eds.), Parallel Symbolic Computing: Languages, Systems, and Applications. Proceedings, 1992. X, 419 pages. 1993.

Vol. 749: P. A. Fritzson (Ed.), Automated and Algorithmic Debugging. Proceedings, 1993. VIII, 369 pages. 1993.

Vol. 750: J. L. Díaz-Herrera (Ed.), Software Engineering Education. Proceedings, 1994. XII, 601 pages. 1994.

Vol. 751: B. Jähne, Spatio-Temporal Image Processing. XII, 208 pages. 1993.

Vol. 752: T. W. Finin, C. K. Nicholas, Y. Yesha (Eds.), Information and Knowledge Management. Proceedings, 1992. VII, 142 pages. 1993.

Vol. 753: L. J. Bass, J. Gornostaev, C. Unger (Eds.), Human-Computer Interaction. Proceedings, 1993. X, 388 pages. 1993.

Vol. 754: H. D. Pfeiffer, T. E. Nagle (Eds.), Conceptual Structures: Theory and Implementation. Proceedings, 1992. IX, 327 pages. 1993. (Subseries LNAI).

Vol. 755: B. Möller, H. Partsch, S. Schuman (Eds.), Formal Program Development. Proceedings. VII, 371 pages. 1993.

Vol. 756: J. Pieprzyk, B. Sadeghiyan, Design of Hashing Algorithms. XV, 194 pages. 1993.

Vol. 757: U. Banerjee, D. Gelernter, A. Nicolau, D. Padua (Eds.), Languages and Compilers for Parallel Computing. Proceedings, 1992. X, 576 pages. 1993.

Vol. 758: M. Teillaud, Towards Dynamic Randomized Algorithms in Computational Geometry. IX, 157 pages. 1993.

Vol. 759: N. R. Adam, B. K. Bhargava (Eds.), Advanced Database Systems. XV, 451 pages. 1993.

Vol. 760: S. Ceri, K. Tanaka, S. Tsur (Eds.), Deductive and Object-Oriented Databases. Proceedings, 1993. XII, 488 pages. 1993.

Vol. 761: R. K. Shyamasundar (Ed.), Foundations of Software Technology and Theoretical Computer Science. Proceedings, 1993. XIV, 456 pages. 1993.

Vol. 762: K. W. Ng, P. Raghavan, N. V. Balasubramanian, F. Y. L. Chin (Eds.), Algorithms and Computation. Proceedings, 1993. XIII, 542 pages. 1993.

Vol. 763: F. Pichler, R. Moreno Díaz (Eds.), Computer Aided Systems Theory – EUROCAST '93. Proceedings, 1993. IX, 451 pages. 1994.

Vol. 764: G. Wagner, Vivid Logic. XII, 148 pages. 1994. (Subseries LNAI).

Vol. 765: T. Helleseth (Ed.), Advances in Cryptology – EUROCRYPT '93. Proceedings, 1993. X, 467 pages. 1994.

Vol. 766: P. R. Van Loocke, The Dynamics of Concepts. XI, 340 pages. 1994. (Subseries LNAI).

Vol. 767: M. Gogolla, An Extended Entity-Relationship Model. X, 136 pages. 1994.

Vol. 768: U. Banerjee, D. Gelernter, A. Nicolau, D. Padua (Eds.), Languages and Compilers for Parallel Computing. Proceedings, 1993. XI, 655 pages. 1994.

Vol. 769: J. L. Nazareth, The Newton-Cauchy Framework. XII, 101 pages. 1994.

Vol. 770: P. Haddawy (Representing Plans Under Uncertainty. X, 129 pages. 1994. (Subseries LNAI).

Vol. 771: G. Tomas, C. W. Ueberhuber, Visualization of Scientific Parallel Programs. XI, 310 pages. 1994.

Vol. 772: B. C. Warboys (Ed.),Software Process Technology. Proceedings, 1994. IX, 275 pages. 1994.

Vol. 773: D. R. Stinson (Ed.), Advances in Cryptology – CRYPTO '93. Proceedings, 1993. X, 492 pages. 1994.

CONTENTS

Cryptosystems

Chair: Scott Vanstone

Stream Ciphers and Cryptographic Functions

Chair: Rainer Rueppel

Proof Systems and Zero-knowledge

Chair: Joan Feigenbaum

Secret Sharing

Chair: Mihir Bellare

Number Theory and Algorithms

Chair: Andrew Odlyzko

Differential Cryptanalysis

Chair: Spyros Magliveras

Complexity Theory

Chair: Joe Kilian

Applications

Chair: Birgit Pfitzmann

Authentication Codes

Chair: Doug Stinson

Hash Functions

Chair: Ivan Damgård

Cryptanalysis

Chair: Eli Biham

Key Distribution

Chair: Tatsuaki Okamoto

Efficient Signature Schemes Based on Birational Permutations

Adi Shamir

Dept. Computer Science, The Weizmann Institute of Science, Rehovot 76100, Israel

Abstract: Many public key cryptographic schemes (such as cubic RSA) are based on low degree polynomials whose inverses are high degree polynomials. These functions are very easy to compute but time consuming to invert even by their legitimate users. To overcome this problem, it is natural to consider the class of birational permutations f over k-tuples of numbers, in which both f and f^{-1} are low degree rational functions. In this paper we develop two new families of birational permutations, and discuss their cryptographic applications.

Remark: At the rump session of CRYPTO 93, Coppersmith Stern and Vaudenay presented two elegant and powerful attacks on the two signature schemes suggested in this paper. The attacks are quite specific, and thus it is conceivable that other signature schemes based on birational permutations will not be affected. The reader is thus encouraged to study the underlying mathematical structure of the schemes and the attacks, but to excercise great caution in implementing the ideas in practice.

1 Introduction

The original proposal for public key cryptography (Diffie and Hellman [1976] was based on the notion of trapdoor permutations, i.e., invertible functions which are easy to compute but apparently difficult to invert, unless some trapdoor information (which makes the inversion easy) is known. The best known implementation of this idea is the RSA scheme (Rivest, Shamir and Adleman [1978]), which can solve in a unified way the problems of key management, secure transmission, user identification, message authentication, and digital signatures. In one of the variants of this scheme, the encryption function is the low degree polynomial $f(x) = x^3 \pmod n$, which can be efficiently computed with two modular multiplications. Unfortunately, its inverse $f^{-1}(v) = v^d \pmod{n}$ is a very high degree polynomial, and thus its evaluation is quite slow (especially in software implementations).

In spite of extensive research in the last 16 years, there had been no fundamentally new constructions of trapdoor permutations which are faster than the RSA scheme. To overcome this difficulty, researchers have developed specialized solutions to various cryptographic needs which are not based on this unifying notion. For example, Diffie and Hellman [1976] proposed a key management scheme which is based on the one-way permutation of exponentiation modulo a prime. Since this function cannot be efficiently inverted, it is neither an encryption nor a signature scheme. The cryptosystem of Merkle and Hellman [1978] is invertible, but its mapping is not onto and thus it can not generate digital signatures. The Fiat-Shamir [1986] and DSS [1991] signature schemes are not one-to-one mappings, and thus they can not be used as cryptosystems.

A natural approach to this problem is to search for low degree algebraic mappings (polynomials or rational functions) whose inverses are also low degree algebraic mappings. Such mappings are called birational functions. We are particularly interested in multivariate mappings $f(x_1, \ldots, x_k) = (v_1, \ldots, v_k)$ in which the x_i and the v_i are numbers modulo a large $n = pq$, since the solution of general algebraic equations of this type is at least as hard as the factorization of the modulus. In this context, we say that a polynomial modulo n is low degree if its degree is a constant which does not grow with n, and a rational function is low degree if it is the ratio of two low degree polynomials. For example, in the case of cubic RSA, the function is considered low degree, but its inverse is not. Nonlinear algebraic mappings do not usually have unique inverses, when they do have inverses they usually cannot be written in closed form, and when the closed forms exist they are usually based on root extractions (radicals) or exponentiations whose computation modulo a large n is very slow. The construction of nonlinear birational mappings is thus a non-trivial task.

One attempt to construct birational permutations, due to Fell and Diffie [85], used the following DES-like idea. Let $(x_1, x_2, \ldots, x_k)$ be an initial k-vector of variables, and let g be a secret nonlinear multivariate polynomial. Alternately replace the current k-vector of multivariate polynomials $(p_1, p_2, \ldots, p_k)$ by $(p_1 + g(p_2, \ldots, p_k), p_2, \ldots, p_k)$, and rotate the k-vector to the right. After sufficiently many iterations, expand and publish the resultant k-vector of multivariate polynomials as your public key. When the trapdoor information g is known, the inverse of f can be computed by undoing the transformations (i.e., by alternately subtracting p_1 and rotating the k-vector to the left). Unfortunately, even when g is a quadratic function, the degree (and thus the size of the public key) grows exponentially with the number of iterations, which cannot be too small for security reasons. As the authors themselves conclude, "there seems to be no way to build such a system that is both secure and has a public key of practical size".

The problem is not accidental, due to the following generic attack: If f is known, the cryptanalyst can prepare a large number of input-output pairs for this function. Since f is invertible, these pairs (in reverse order) can be used to interpolate the unknown low-degree function f^{-1} by solving a small number of linear equations relating its coefficients. We do not know how to solve this problem in the context of public key cryptosystems. However, in the context of public key signature schemes there is a simple way to avoid this specific attack with the following modification:

Key Generation: Each user in the system chooses a particular birational permutation $f(x_1,\ldots,x_k)=(v_1,\ldots,v_k)$ consisting of k rational functions $f_i(x_1,\ldots,$ $x_k) = v_i$, discards the first $s > 0$ of these f_i functions, describes the other $k - s$ f_i functions in his public key, and keeps the inverse of f as his private key.

Signature Generation: Given a digital message m, the signer chooses $v_i = r_i$ for $i = 1,\ldots,s$, and computes $v_i = h(m,i)$ for $i = s+1,\ldots,k$, where r_i are newly chosen secret random values, and h is a publicly known cryptographic hash function. He then uses his knowledge of the secret f^{-1} to compute a signature $(x_1,\ldots,x_k)$ satisfying $f(x_1,\ldots,x_k) = (v_1,\ldots,v_k)$.

Signature Verification: The verifier checks that $f_i(x_1,\ldots,x_k) = h(m,i)$ for $i = s+1,\ldots,k$, where the f_i's are taken from the signer's public key.

This modified scheme can no longer be used as a cryptosystem, since the cleartext $(x_1,\ldots,x_k)$ cannot be uniquely recovered from the shorter ciphertext $(v_{s+1},\ldots,$ $v_k)$. It can be used as a signature scheme, since messages can have multiple signatures. The cryptanalyst cannot interpolate f^{-1} since it is not uniquely defined by the public key: He cannot generate complete input-output pairs for f^{-1} by himself, and cannot use input-output pairs generated by the legitimate signer since each one of them is based on new unknown values r_i. The recommended choice of s is 1, which makes the verification condition hardest to satisfy.

The security of this scheme depends on the choice of birational permutations. In Section 2 we introduce a simple family of birational permutations based on sequentially linearized equations, and in Section 3 we introduce a more sophisticated family of birational permutations based on algebraic bases in polynomial rings. Both families yield signature schemes with very low computational complexity. Unfortunately, both families also turn out to be breakable by the new attacks of Coppersmith Stern and Vaudenay[1993]. The reader is encouraged to look for other families of birational permutations and to study their efficiency and security.

2 A Birational Permutation Based on Sequentially Linearized Equations

Let n be the public product of two large secret primes p and q. verify Consider the triangular birational permutation $g(y_1,\ldots,y_k) = (w_1,\ldots,w_k) \pmod n$ in which the i-th output depends only on the first i inputs via the mapping $g_i(y_1,...,y_i) = w_i \pmod n$, where g_i is a low degree polynomial which is linear in its last input y_i (the other inputs can and should occur non-linearly). Given the values of inputs $y_1,\ldots,y_k$, we can easily compute the values of the outputs $w_1,\ldots,w_k$ by evaluating k low degree polynomials. Given the values of the outputs $w_1,\ldots,w_k$, we can easily recover the values of the inputs $y_1,\ldots,y_k$ by solving a series of linear equations: First we solve for y_1 in $g_1(y_1) = w_1 \pmod n$. Then we substitute the computed value of y_1 into its (non-linear) occurrences in $g_2(y_1,y_2) = w_2 \pmod n$, and solve the remaining linear equation in y_2. We proceed in this order until we compute the last y_k. Each y_i is thus a low degree rational function of the w_j's, which is easy to compute with a small number of arithmetic operations modulo n.

To hide the easy solvability of the g_i's, the user has to transform them into more random looking polynomials before publishing them as his public key. We recommend the following two transformations:

1. Let A be a randomly chosen invertible $k \times k$ matrix, and consider the variable transformation $Y = AX$, where Y is the column vector of original variables $(y_1,\ldots,y_k)^t$ and X is a column vector of new variables $(x_1,\ldots,x_k)^t$. When the resultant polynomials are expanded, they contain all the variables in a non-linear way.
2. Let B be a randomly chosen invertible $k \times k$ matrix, and consider the mixing transformation $F = BG$, where G is the column vector of polynomials $(g_1,\ldots,g_k)^t$ and F is a column vector of new polynomials $(f_1,\ldots,f_k)^t$. Each f_i is thus a polynomial whose coefficients are random linear combinations of the corresponding coefficients of the given $g_1,\ldots,g_k$.

When A and B are known, it is easy to solve the resultant system of equations $f_i(x_1,...,x_k) = v_i \pmod n$ for $i = 1,...,k$ by inverting these transformations. of them change be To minimize the size of the public key, we recommend using g_i's which are homogeneous quadratic expressions of the form:

$$g_i(y_1,\ldots,y_i) = l_i(y_1,\ldots,y_{i-1}) \cdot y_i + q_i(y_1,\ldots,y_{i-1}) \pmod n$$

where l_i is a randomly chosen linear function of its inputs and q_i is a randomly chosen homogeneous quadratic function of its inputs. The only exception is g_1 in which l_1 and q_1 have no inputs. Since the coefficients of the linear g_1 cannot be mixed with the coefficients of the quadratic $g_2, \ldots, g_k$, and since we have to eliminate at least one of the polynomials in order to overcome the interpolation attack, we recommend the elimination of g_1 from the user's public key.

Without loss of generality, we can assume that $g_1(y_1) = y_1$ and $g_2(y_1, y_2) = y_1 y_2$ (since they can always be brought to this form by linear transformations). The case $k = 2$ is thus equivalent to the OSS scheme (Ong, Schnorr, Shamir [STOC83]), where the variable transformation A is $y_1 = x_1 + ax_2$, $y_2 = x_1 - ax_2$ and the mixing transformation B is the identity (all the arithmetic operations are carried out modulo a composite n). solves the signature The OSS scheme was successfully attacked by Pollard [1984], who showed that one quadratic equation in two variables can be solved even when the factorization of the modulus is unknown. A typical example of the extended signature scheme for $k = 3$ with the toy modulus $n = 101$ is:

Example: Consider the following sequentially linearized system of equations:

$$\begin{aligned} y_1 &= w_1 \pmod{101} \\ y_1 y_2 &= w_2 \pmod{101} \\ (29y_1 + 43y_2)y_3 + (71y_1^2 + 53y_2^2 + 89y_1y_2) &= w_3 \pmod{101}. \end{aligned}$$

Apply the linear change of variables:

$$\begin{aligned} y_1 &= x_1 + 25x_2 + 73x_3 \pmod{101} \\ y_2 &= x_1 + 47x_2 + 11x_3 \pmod{101} \\ y_3 &= x_1 + 83x_2 + 17x_3 \pmod{101} \end{aligned}$$

to obtain the new expressions:

$$\begin{aligned} x_1 + 25x_2 + 73x_3 &= w_1 \pmod{101} \\ x_1^2 + 64x_2^2 + 96x_3^2 + 72x_1x_2 + 84x_1x_3 + 70x_2x_3 &= w_2 \pmod{101} \\ 83x_1^2 + 55x_2^2 + 16x_3^2 + 28x_1x_2 + 97x_1x_3 + 74x_2x_3 &= w_3 \pmod{101}. \end{aligned}$$

Mix these three expressions g_1, g_2, and g_3 by computing: $f_1 = g_1 \pmod{101}$, $f_2 = (39g_2 + 82g3) \pmod{101}$, $f_3 = (93g_2 + 51g3) \pmod{101}$. The resultant expressions are:

$$\begin{aligned} x_1 + 25x_2 + 73x_3 &= v_1 \pmod{101} \\ 78x_1^2 + 37x_2^2 + 6x_3^2 + 54x_1x_2 + 19x_1x_3 + 11x_2x_3 &= v_2 \pmod{101} \\ 84x_1^2 + 71x_2^2 + 48x_3^2 + 44x_1x_2 + 33x_1x_3 + 83x_2x_3 &= v_3 \pmod{101}. \end{aligned}$$