Complex Systems **1** (1987) 51–57

Cellular Automaton Public-Key Cryptosystem

Puhua Guan
Department of Mathematics, University of Puerto Rico, Rio Piedras, PR 00931, USA

Abstract. A public-key cryptosystem based on inhomogeneous cellular automata is proposed. The running time of all known algorithms for breaking the system grows exponentially with the cipher block length.

1. Introduction

A cryptographic system is a mathematical system for encrypting or transforming information so that it appears useless to those who are not meant to have access to it. Any cryptographic technique, such as the substitution and transposition of symbols, that operates on a message without regard to its linguistic structure is called a cipher and is said to generate a cipher text. In a public-key cryptosystem, a receiver, rather than agreeing with each sender on how to operate on a message, simply generates two distinct keys of his own: an enciphering key E, which is communicated to the public, serves the purpose of encryption; a deciphering key E, which is kept by the receiver himself, serves to implement the system's deciphering algorithm.

A cryptosystem should satisfy the following three requirements:

Security. For people who do not know the deciphering key, it should require an unrealistic amount of time to recover the plain information from a cipher text, whereas for the receiver who knows the deciphering key, the original information should be quickly recoverable from the cipher text.

Integrity. If an enemy agent attempts to confuse the receiver by inserting a corrupted message, the receiver should be able to detect it.

Authorization. If someone sends a message using another person's key, the receiver should be able to detect it.

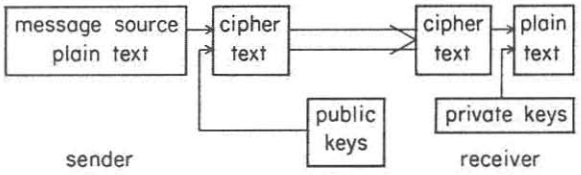

Figure 1: Schematic arrangement of a public-key cryptosystem.

Several different public-key cryptosystems have been proposed [1]; many techniques for attacking them have also been developed [3]. This paper presents a new public-key cryptosystem that satisfies all the requirements previously mentioned. The running times of all algorithms so far known for breaking our system grow exponentially with cipher block length.

Section 2 describes our system from the point of view of the security requirement. The end of the section shows how the integrity and the authorization requirements can also be satisfied. Section 3 provides an example.

2. Cellular automaton cryptosystem

To transform information by electronic means, a message is usually represented by a string of binary bits. This representation is always done according to some well known rules (ASCII, etc.).

This string of binary bits is called the plain text. The string of binary bits is then cut into blocks, with each block containing a certain number of bits. The bits in a block can be viewed as an element of another set S, for example, every three bits represent an element in a set of size 8, etc. These elements of S will be the fundamental units of our cryptosystem. Let S be the ground set of the system. The blocks are the independent units of the cryptosystem; in the cipher text, each block is a replacement of a block in the plain text.

Suppose each block is N bits long, representing m elements of S. In order to meet the security requirement, an invertible function is needed that maps S^m to S^m and satisfies the following conditions:

(a) It is easy to compute (for enciphering).

(b) It is hard to find its inverse (for deciphering by intruders).

(c) With some key information, the inverse image can be easily computed.

The complexity of behavior seen in cellular automata suggests that invertible cellular automaton rules are promising candidates for our purpose. However, cellular automaton rules are usually represented by tables, and if the effective neighborhood size is small, then there is a danger that the inverse image in any particular case can be found by a random attack. If

the effective neighbourhood size is large, then the table will be too large, since its size grows exponentially with the number of neighbours.

To make rules which can be stated succinctly but which have large effective neighborhood sizes, each S is associated with a mathematical structure. For example, we can think of S as a mathematical ring or a field and use multivariate polynomials to represent the cellular automata rules. Note that when $|S|$ is a prime power, then every multivariate function over S is a polynomial function. When $|S|$ is not a prime power, a large portion of multivariate functions can still be represented as polynomial functions. When the degree of each polynomial function is bounded by a small number d, then the size of each polynomial is bounded by m^d, where m is the number of the variables. So we can have cellular automata rules with large effective neighborhood sizes but with short representations. On the other hand, multivariate polynomial functions satisfy conditions *(a)* and *(b)* well, since polynomials are easy to compute. But the time needed to solve a system of nonlinear polynomial equations in general grows exponentially wth the number of variables [3,4,7].

To obtain a system that satisfies all *(a)*, *(b)*, and *(c)*, we first make the following definitions.

Definition 1. *A finite cellular automaton of size m is a dynamical system with m sites $(x_1^t, x_2^t, \ldots, x_m^t) = x^t$, together with a set of mappings $\{F_i^t\}$ at each discrete time t, such that*

$$x_i^{t+1} = F_i(x_1^t, x_2^t, \ldots, x_m^t), \tag{2.1}$$

where x assumes values in any set S and the subscripts are calculated modulo m.

Following the above definitions, if $F_i^t = F_j^t$ for each i, j, then the cellular automaton is homogeneous, and if there exists $i \neq j$ such that $F_i^t \neq F_j^t$, then the cellular automaton is inhomogeneous. If $F_i^t = F_i^s$ for all t, s and i, then the cellular automaton is time stable, and if there exists t and s such that $F_i^t \neq F_i^s$, then the cellular automaton is time varying.

Not much investigation has been done on time varying or inhomogeneous rules. For the time stable and homogeneous rules, the behavior of most cellular automata appears unpredictable [6]. Complete descriptions have so far been found only for additive cellular automata.

Definition 2. *A cellular automaton is partially linear at the time t if S is a ring and some F_i^t are linear functions. It is partially linear invertible if the coefficients of those linear functions together form an invertible matrix.*

Definition 3. *A cellular automaton is s-fold linear invertible at the time t if S is a ring and the variables $(x_1^t, x_2^t, \ldots, x_m^t)$ can be partitioned into s parts $(x_{11}^t, \ldots, x_{1k_1}^t), \ldots, (x_{s1}^t, \ldots, x_{sk_s}^t)$ such that for each j, there are exactly k_j*

functions in the set $\{F_i^t\}$ that are linear functions in the variables of the j-th part. The functions that are linear in the variable of the j-th part can be any functions of the variables in the previous parts. Moreover, the variables of the latter parts can not appear in these functions, and the coefficients of the variables of the j-th part in these functions form an invertible matrix.

Two examples of 2-fold linear invertible cellular automata are given in Section 3.

For any system of size m with an initial state $(x_1, x_2, \ldots, x_m)$, we can easily compute the state at the next time $(x_1', \ldots, x_m')$ under any s-fold linear invertible rule. It is also easy to trace back $(x_1, \ldots, x_m)$ from $(x_1', \ldots, x_m')$. However, if we compose several multifold linear invertible rules together, the composite function is no longer partially linear. To find the original state from the final state obtained by the action of the composed rules, it is then necessary to solve a system of nonlinear polynomial equations, if one knows only the composite function. On the other hand, the designer of the rules, knowing how the composite function is constructed, can give a procedure for recovering the inital values without solving general equations.

Now our cryptographic scheme is clear. Let the ground set be a commutative ring. The enciphering key E is a composition of several time-varying inhomogeneous multifold linear invertible rules, which is made public. The deciphering key D, which is kept private by the designer, is the set of the individual rules in the composite enciphering function.

The requirements of integrity and authority can be satisfied as follows. After the sender sends the cipher text M including his own name enciphered according to the public key of a receiver, he applies the inverse of his own public key to M, gets M', then sends M' as well. The receiver first deciphers M and finds the sender's name, then applies the public key of the sender to M'. If he gets M as given in the first half of the cipher text, he can believe that the signature is authentic and the information not coded by an enemy agent.

3. Examples

Suppose we have a system with a block length of 5 bits and assume that each bit takes a value in the field of 2 elements. A user C publishes the following public keys:

$$\begin{aligned} y_1 &= x_1x_2 + x_5 \\ y_2 &= x_2x_3 + x_4 \\ y_3 &= x_1x_2x_3 + x_1x_2x_4 + x_2x_3x_5 + x_4x_5 + x_2 \\ y_4 &= x_2x_1 + x_2x_5 + x_3 \\ y_5 &= x_1 + x_2 \end{aligned} \tag{3.1}$$

If B wants to send C the message 10110, he first looks up the above rule under C's name and applies the rule to 10110, then sends 01011.

The rule is actually composed of

$$\begin{aligned} x_1' &= x_2 \\ x_2' &= x_3 \\ x_3' &= x_1 \\ x_4' &= x_5 + x_1x_2 \\ x_5' &= x_4 + x_2x_3 \end{aligned} \tag{3.2}$$

and

$$\begin{aligned} y_1 &= x_4' \\ y_2 &= x_5' \\ y_3 &= x_1' + x_4'x_5' \\ y_4 &= x_2' + x_1'x_4' \\ y_5 &= x_3' + x_1'. \end{aligned} \tag{3.3}$$

C keeps (3.2) and (3.3) to himself. Upon receiving 01011 he can solve (3.3), to get 01101 for x_i and then solve (3.2), to get 10110.

In general, if the length of the block is n bits, and to represent an element of the ground set needs k bits, then the size of the keys is bounded by $(\frac{n}{k})^d$, where d is the maximum degree of the keys. In fact, we can choose d to be as small as 2 or 3. Known algorithms for solving systems of nonlinear system of equations take an expected time $O(2^n)$. In particular, when the ground set is the field of two elements, the general problem of solving a nonlinear system of equations known to be NP complete [8].

Acknowledgements

I am grateful to Professors Wolfram and Zassenhaus for their valuable comments and suggestions for the improvement of this paper.

References

[1] Martin E. Hellman, "An Overview of Public Key Cryptography", *IEEE Transactions on Communications*, **16** (1978)

[2] R. Rivest, A. Shamir, and L. Adleman, "A Method For Obtaining Digital Signatures and Public Key Cryptosystems", *Communications of the ACM*, **21** (1978) 120-126.

[3] R. Blakely and G. Blakely, "Security Of Number Theoretic Public Key Cryptosystems Against Random Attack I, II, III", *Cryptologia*, **2** (1978) 305-321; **3** (1979) 29-42; **3** (1979) 105-118.

[4] G. E. Collins, "Quantifier Elimination For Real Closed Field: A Guide To the Literature", *Computer Algebra*, edited by B. Buchberger, G. E. Collins, R. Loos, (Springer-Verlag, NY, 1982), 79-81.

[5] M. J. Fisher, and M. O. Rabin, "Super Exponential Complexity of Presburger Arithmetic", in *MIT MAC Technical Memo 43.*

[6] Stephen Wolfram (Editor), *Theory and Applications of Cellular Automata*, (World Scientific, 1986).

[7] Puhua Guan, "Analysis Of Cellular Automata Public Key Cryptography", submitted to 1987 Symposium on Theory of Computing.

[8] Puhua Guan and H. Zassenhaus, "Solving Systems of Equations Over Finite Fields", (to be published in *Journal of Number Theory*, February 1987).

[9] Puhua Guan, "Public-Key Cryptosystem Based On Higher Order Cellular Automata", submitted to *IEEE Transactions on Information Theory*, 1987.