CELLULAR AUTOMATA BASED KEY AGREEMENT

Debdeep Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur, India

Dipanwita RoyChowdhury
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur, India

Keywords: Cellular Automata, Key Agreement, Random Oracles, one-way function.

Abstract: This paper proposes a new key agreement protocol using Cellular Automata (CA). The primitives on which the protocol is based has been developed using specially designed CA structures. The paper aims at developing a key agreement technique which is not based on the Diffie-Hellman problem. The removal of exponentiations makes the protocol fast and have a linear time complexity. The protocol has been found to resist known forms of attacks. Indeed the initial review promises the development of a key agreement protocol which meets nicely the conflicting ends of security and efficiency.

1 INTRODUCTION

Authentication and key establishment are fundamental building blocks for securing electronic communication. Cryptographic algorithms, for encryption and decryption cannot perform their specified functions unless secure keys have been established. Key establishment (A.Menezes and Vanstone, 1996) is a process or protocol whereby a shared secret becomes available to two or more parties, for subsequent cryptographic use. Key agreement mechanism is a key establishment technique in which a shared secret is derived by two or more parties as a function of information contributed by or associated with each of these (ideally) such that no party can predetermine the resulting value. Since the seminal paper of Diffie and Hellman (W.Diffie and M.Hellman, 1976) in 1976, several solutions have been proposed for key agreement whose security lies on the Diffie-Hellman problem (either computational or decisional) in finite groups. However modular exponentiation is an expensive operation. Despite the increase in availability of computational resources considerable demand exists in the development of protocols that can be implemented on devices with limited computational power. Applications of such type being mobile handsets, handheld devices, embedded hardware and smart cards.

In this paper we propose a Cellular Automata (CA) based key agreement protocol. Cellular Automaton has a regular, cascadable and modular structure thus giving scope to efficient implementation(Chaudhuri et al., 1997). The paper shows a CA based technique to implement a k-bit to k-bit trapdoor permutation f which is used to build an encryption function which is secured under the Random Oracle (RO) model (Mihir Bellare and Phillip Rogaway, 1994). The encryption function is used in combination with a specially designed Cellular Automaton called Two Predecessor Single Attractor Cellular Automata (TPSA-CA) to derive a common secret key K_{AB}. Complemented Cellular Automaton has been characterized in (Mukhopadhyay and RoyChowdhury, 2004). This characterization has been extended in the present paper where the complemented CA has been used to develop a technique to vary the key at a very fast rate.

As a matter of general principle it is not possible to establish an authenticated session key without existing secure channel already being available. In fact this statement has been stated formally and proved to be correct(Boyd, 1993). The proposed protocol also assumes the existence of an offline server which provides certified long term keys to the principals. It may be noted that these long term keys are used to encrypt random bit strings and are also used only in the key-establishment phase. But the resultant of the agreement protocol is a session-key which is used to encrypt files of fixed formats (like bmp files or jpeg files). Also they are used for bulk encryption and so are more amenable to cryptanalysis. Thus they

Mukhopadhyay D. and RoyChowdhury D. (2005).
CELLULAR AUTOMATA BASED KEY AGREEMENT.
In *Proceedings of the Second International Conference on e-Business and Telecommunication Networks*, pages 262-267
DOI: 10.5220/0001412402620267

are required to be altered frequently(Boyd and Mathuria, 2003). The complemented CA rules and the one way function provide a way to vary the key frequently which is presented in the paper. The proposed protocol has been shown to be secured in the face of known attacks. The removal of the exponentiation operation leads to huge computational savings. Keys can be guaranteed to be random even if one of the inputs become known - a property lacking in D-H based protocols(Boyd and Mathuria, 2003).

The outline of the paper is as follows: *Section 2* develops the CA based primitives for the key agreement protocol. *Section 3* proposes the key agreement protocol using the CA based primitives. *Section 4* discusses the security of the scheme and the work is concluded in *section 5*.

2 CA BASED PRIMITIVES FOR THE KEY AGREEMENT PROTOCOL

Key agreement protocols require the strength of their underlying primitives. The basic components (primitives) on which the present agreement protocol depends are enumerated below:

1. A CA based Encryption function
2. A CA based Agreement function which uses the information contributed by the two principles
3. A CA based function for varying the session key at a very fast rate

In the following subsections the cryptographic primitives are explained.

2.1 CA based encryption function

In (Mihir Bellare and Phillip Rogaway, 1994) Bellare and Rogaway have proved that under the Random Oracle (RO) model the following encryption is semantically secured.

$$E^{G,H}(x) = f(x + G(r)||r + H(x + G(r))$$

Here, $+$ indicates the bitwise exor operation and $||$ indicates the string concatenation. The message to be encrypted is a k bit binary sequence and r is a random k bit number. In the above expression H and G are random oracles and f is a trapdoor permutation. In the RO model it is assumed that H and G are functions truly returning random values each time, except that it always return the same output when the input is same. They are a k bits $\rightarrow$ k bits pseudo-random permutations. In our construction f is chosen as a CA based one-way permutation which operates on $2k$ bits and produces a $2k$ bits output. The functions H and G are realised using CA based structures which are known to be excellent pseudo-random number generators(Chaudhuri et al., 1997; Hortensius et al., 1989).

Security is based on the reasoning that if an adversary A can break the encryption scheme with some success probability, then there is an algorithm M which can invert the underlying trapdoor permutation f with comparable probability and time. So, first we outline the CA based one way function, which is based on simple rules, thus leading to efficient implementations.

2.1.1 One Way function based on Simple Rules

A function f is one-way if

1. It is polynomial time computable
2. It is hard to invert i,e no polynomial time algorithm can find any pre-image of $f(x)$.

Let the input of the one way function be $X = \{x^{n-1}, x^{n-2}, \ldots, x^0\}$ and the output is $Y = \{y^{n-1}, y^{n-2}, \ldots, y^0\}$.

We define a state transition by the following function:

$$\overline{y^n} = x^{n+1} + (x^n \bigvee x^{n-1})$$

Here, the bar symbol indicates the bitwise complement, $\bigvee$ indicates the bitwise or operator and $+$ indicates the bitwise exor operator. The cellular automaton has a zero boundary condition, i,e $x^{-1} = x^n = 0$. Thus the output Y can be easily constructed from X but the inverse is hard. A mapping $x \rightarrow \phi(x)$ is called invertible if $\phi(x) = \phi(y)$ iff $x = y$. According to that definition the present function is not invertible.

However, the function (state transition) can be easily inverted if the value of x^0 is known. Thus depending on x^0 there can be two values (pre-images) of Y. If such a transformation is used 2 times any output state can have 2^2 pre-images with equal probability if the last bits of the transformation are random to an adversary. Thus if the transformation is repeated n number of times there can be 2^n pre-images with equal probability. Inorder to decrypt we have a key r (which is a long term key in our algorithm). The bits of r are exored to the 0^{th} bit slices of all the stages of the transformation. The information is sent as a part of the encrypted message. The part of the message is called the trap-pad (denoted by P). The application of the above transition one time is denoted by $N(X)$ and for n times by $N^n(X)$. Thus we construct the one-way function with the trapdoor as follows:

$$Y = N^n(X) = N^n(x^{n-1}, x^{n-2}, \ldots, x^0)$$

The 0^{th} bit slices at all the stages of the transformation are stored in an n bit register. Thus $P = ([N(X)]_0 + r_0, [N^2(X)]_0 + r_1, \ldots, [N^{n-1}(X)]_0 + r_{n-1})$. Thus the one way permutation is denoted by

$$\{P, Y\} = \{P, N^n(X)\}$$

Thus any principal (party) with a valid long term key (r) can easily decrypt the sequence and obtain X while for an adversary all the pre-images are equally probable.

2.2 Two Predecessor Single Attractor Cellular Automata (TPSA-CA) Based Function

TPSA CA are a special class of non-group CA in which the state transition graph forms a single inverted binary routed tree at all zero state. The CA has been characterized in (Chaudhuri et al., 1997) and has been used in the current protocol to develop an efficient technique to agree on the key for the first time. These CA are characterized by the fact that every reachable state in the state transition graph has exactly two predecessors. The only cyclic state is the all zero state (for a non-complemented TPSA CA), which is an attractor (or graveyard). Corresponding to a TPSA CA M_1 and a state S, there exists a complemented CA M_2 with state S as an attractor. If the characteristic matrix M_1 be indicated by T_p and it is required to build a complemented TPSA CA such that S is the graveyard (attractor) then the characteristic matrix of the complemented CA, T_c is related to T_p by

$$T_c(X) = T_p(X) + (I + T_p)Z$$

where X is the seed to the CA.

2.2.1 How TPSA CA provides key freshness?

The state transition graph of an n bit TPSA Cellular Automaton is shown in figure 1. Given two inputs A and B the path between the points are noted. The path between A and B, with both ends included give a list of indices that forms a set. For a given TPSA CA we have a unique path between A and B. However the path depends on the graveyard state, which can be changed as already mentioned. The elements of the path is used to return the modulo-2 sum of the selected numbers which form elements of the list. Since the state transition depends on the graveyard state and all the states can be made the graveyard, all the paths and final sum are possible for a given pair of A and B. It may be noted that for an n bit TPSA Cellular Automaton the path between A and B has atmost $2n$ elements i,e twice the depth of the inverted binary tree. So, the computation of the output sum can take place in linear time (O(n)).

The final sum of the path is the agreed key K_{AB} which is the output of the contributions of the two parties, say N_A and N_B. Mathematically, $K_{AB} = TPSA(N_A, N_B)$, where $TPSA$ is the TPSA CA based function explained. Since the transition characteristic T of the TPSA CA is not invertible (note the graph) if the final path sum and one of the inputs say N_A is known then it is not possible to calculate the other element N_B without exhaustive search. This prevents a principal B to calculate the value of N_B so that he can force the value of N_A and N_B to output an old key (old sum). Such type of function provides key-freshness to the key.

2.3 CA based function for varying the key

In this subsection we provide the properties of a special type of complemented Cellular Automaton which is used to vary the key at a fast rate.

One of the rules of Cellular Automata ($\overline{T}$) is rule 153, (Chaudhuri et al., 1997). The present section characterizes the CA with rule 153. It is known that if a cellular automaton with rule 153 is fed with an initial seed of X, then the cellular automaton produces an output $\overline{T}(X) = T(X) + IF$, where I is a unit matrix and F is all one vector. Hence, we have $X, \overline{T}(X)$ and $\overline{T}^2(X)$ members of the same cycle. Physically, an n-cell uniform CA having rule 153 evolves with equal number of cyclic states. The CA has some remarkable properties. The CA evolves equal lengths and the length for an n-cell CA grows linearly with the number of cells.

The following theorem characterizes a CA based on the rule $\overline{T}$.

Theorem 1 *(Mukhopadhyay and RoyChowdhury, 2004) The length of cycle for an n-cell CA, having rule $\overline{T}$, is*

$$l = \ 2^{\lfloor logn \rfloor + 1} \quad , n \geq 2 \tag{1}$$

2.3.1 Generalisation of 153 CA

The above characterization for the Cellular Automaton is for rule 153. Similarly, it may be proved that the same characterization holds for generalised Cellular Automata with characteristic matrix, T_g where $T_g = A.(\overline{T}).A^{-1}$. In order to relate the state spaces of the generalised CA, we define two rules:

$$R1(X) = X + \overline{T}_g(X) + \overline{T}_g^2(X)$$

$$and, R2(X) = X + \overline{T}_g(X) + \overline{T}_g^3(X)$$

It may be proved that for a and b randomly chosen indices, $\overline{T}_g^a R1R2(\overline{T}_g^b(X)) = \overline{T}_g^b R2R1(\overline{T}_g^a(X))$. Every element, X of a cyclic subspace when mapped by the rule $R1$ (or $R2$) gives another cycle. The rules or the properties of the CA are used to migrate from one cyclic subspace generated by T_g to another. The inter-relations promise the development of CA-based algorithms used to develop an agreement property, which is illustrated in figure 2.

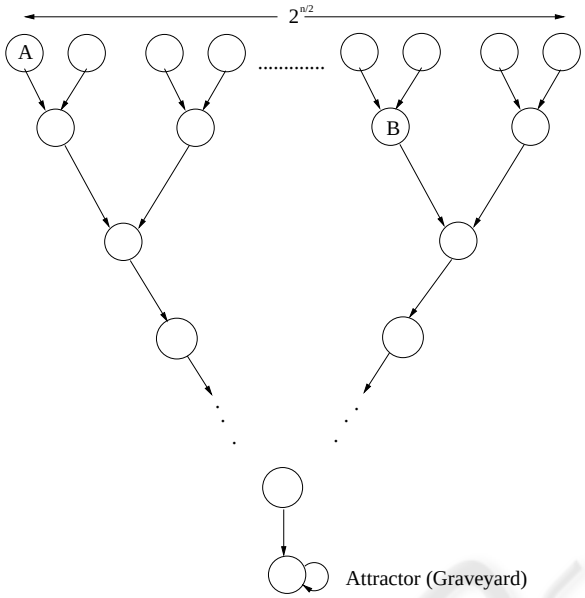

Figure 1: The state transition graph of a TPSA CA

2.3.2 Regeneneration of session keys

Key freshness is important, that is the key should change frequently. Thus less data is encrypted by the same key, so that the eavesdropper has lesser probability of success. Further the amount of damage done due to the revelation of a key is reduced if the key is frequently changed.

The class of the group CA characterized with the matrix T_g exhibit an interesting agreement property. Figure 2 shows the state space cycles. P1 is an initial point of agreement of two parties. After the initial agreement both takes up different paths (as shown by the dashed and the solid lines). It is certain from the above results both the paths converge again at P2 which is the second point of collision or agreement. This property promises the development of an efficient agreement protocol using Cellular Automata.

3 ESSENTIALS OF THE PROTOCOL

The Cellular Automata based key agreement protocol is presented in this section. The two parties A and B exchange messages to derive a final key K_{AB}.

Message 1: $A \rightarrow B$, $E_B(A, N_A, Z)$
Message 2: $A \leftarrow B$, $E_A(B, N_B), x_B$
Message 3: $A \rightarrow B$, x_A
The terms used in the protocol are explained next.

- Principal A uses the CA based encryption scheme (proposed in section 2.1) to encrypt the identifier A i,e E_A, the nonce N_A and the secret graveyard state Z on which the state transition graph of the TPSA CA is dependent. The encrypted text has a trappad P which in combination with B's long term key is used by B to decrypt the ciphertext and obtain the secret graveyard (explained in section 2.1.1).
- B then sends his contribution N_B to A using the CA based encryption function (i,e E_B). Meanwhile, B computes the shared key K_{AB} by using the TPSA CA based on the secret graveyard state. The values of N_A and N_B are as outlined in section 2.2. B also generates a random number r_B and computes a value $x_B = R2[T_g^{r_B}(K_{AB}+f(K_{AB})]$, where f is the CA based one way function and T_g is the generalised Cellular Automaton characterized in section 2.3.2. B then sends back x_B to A who also uses the values of N_A, N_B and Z to compute the value of K_{AB}.
- A also generates a random number r_A and calculates $x_A = R2[T_g^{r_A}(K_{AB} + f(K_{AB})]$. The value of x_A is then sent to B.

Using the following theorem (agreement property) both A and B can continnue to generate new keys at a very fast rate.

Theorem 2 *A computes K_1 from the knowledge of x_B (the public information), r_A (the private information) and the initially agreed key K_{AB}. B computes K_2 from the knowledge of x_A (the public information), r_B (the private information) and the initially agreed key K_{AB}. Then $K_1 = K_2$, thus they both agree on a new derived key.*

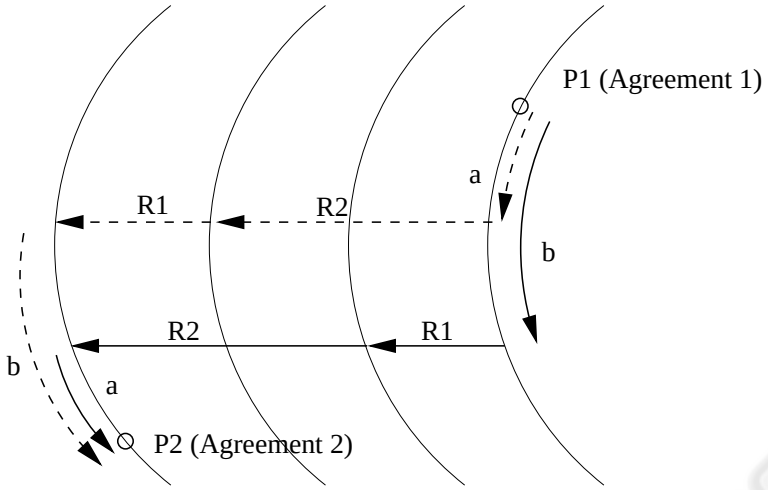

Figure 2: Agreement Property of the State Spaces

Proof: The proof follows directly from the characterization presented in the previous section. A calculates $K_1 = T_g^{r_A}[R1\{R2[T_g^{r_B}(K_{AB} + f(K_{AB}))\}]$ and B computes $K_2 = T_g^{r_B}[R2\{R1[T_g^{r_A}(K_{AB} + f(K_{AB}))\}]$. From the property of the generalised automata, $K_1 = K_2$. Thus they both agree on a new key. □

4 SECURITY ASPECTS OF THE PROTOCOL

The key agreement protocol's security is dependent on the security of CA based encryption and the pseudo-randomness of the CA based functions x_A and x_B (generated using T_g).

The CA based encryption is secured under the Random Oracle method (Mihir Bellare and Phillip Rogaway, 1994). The security of the encryption is based on the security of the underlying CA based one way function. For an adversry who does not have the secret information (long-term key) all the elements can be preimages of the output $f(x)$ with equal probability.

A TPSA CA based function is used to derive the agreed key K_{AB}. Since the graveyard state Z is kept secret from the adversary this function helps in deriving the key with freshness property (explained in subsection 2.2).

Finally the pseudo-randomness of x_A and x_B has been verified using standard tests for randomness. In figure 3 the results of the avalanche analysis has been presented. The Avalanche Test is based upon the fact that a 1-bit change in the plain text should produce a radical change in the ciphered text (Stallings, 2003). A pair of input sequences have been taken and the function is applied upon both of them. The number of bit-differences in the output sequence are noted. The mean of the distribution should be n/2 and the standard deviation, $\sqrt{npq}$, where n is the block-size and p=q=1/2 (binomial probability). The key-agreement protocol has been simulated on large number of data and derived keys have been observed pairwise. The results have been found to comply with the expected.

The randomness of the values x_A and x_B can also be proved if it is assumed that r_A and r_B are random. The randomness of the values are further enhanced by the use of the CA based one way function which is a slight variant of rule 30 CA (which is known to be a powerful random sequence generator (Wolfram, 1983). Also the generalisation of the 153 CA shows how the randomness can be varied by constructing various T_g's where the characterization and agreement properties are still valid. It has been checked that the protocol prevents known attacks like replay attacks, known key attacks and triangle attacks.

5 CONCLUSION

The paper proposes a Cellular Automata based key agreement protocol. The protocol is based on CA based primitives which have been developed in the paper. The security of the key agreement protocol is based on the security of a CA based encryption scheme which is secured under the Random Oracle Method. Since the protocol does not have any exponentiation operation and is based on simple rules the technique can be used to generate session keys at a very fast rate. It has been checked that the protocol prevents known attacks like replay attacks, known key

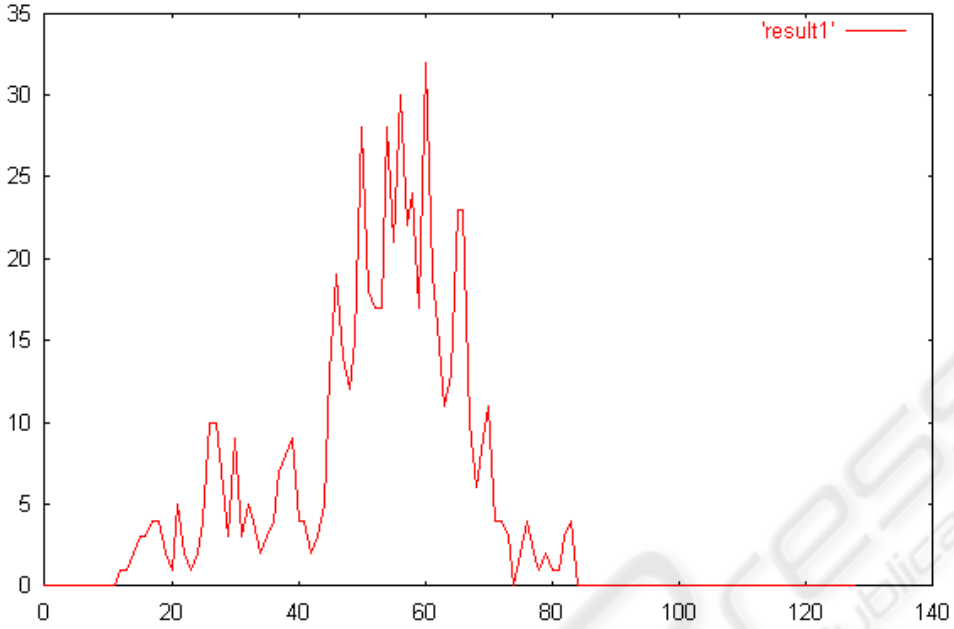

Figure 3: Avalancle Test Results

attacks and triangle attacks. However the history of agreement protocols obviates the necessity of formal proofs of such protocols and is currently underway.

REFERENCES

A.Menezes, P. V. O. and Vanstone, S. (1996). *Handbook of Applied Cryptography*, chapter 12, pages 489–541. CRC Press.

Boyd, C. (1993). Hidden assumptions in cryptographic protocols. *IEEE Journal of Selected Areas in Communications*, 11(5):694–701.

Boyd, C. and Mathuria, A. (2003). *Protocols for Authentication and Key Establishment*, chapter 5, pages 138–199. Springer.

Chaudhuri, P. P., Chowdhury, D., Nandi, S., and Chattopadhyay, S. (1997). *Additive Cellular Automata Theory and its Application*, volume 1, chapter 4. IEEE Computer Society Press.

Hortensius, P. D. et al. (1989). Cellular automata based pseudo-random number generators for built-in self-test. 8(8):842–859.

Mihir Bellare and Phillip Rogaway (1994). Optimal Assymetric Encryption-How to Encrypt with RSA. In *Advances in Cryptology-Eurocrypt 1994*. Lecture Notes in Computer Science, Vol. 950, Springer Verlag.

Mukhopadhyay, D. and RoyChowdhury, D. (2004). Characterization of a class of complemented group cellular automata. In *In the Proceedings of ACRI 2004, LNCS 3305*. University of Amsterdam, Science Park Amsterdam, The Netherlands.

Stallings, W. (2003). *Cryptography and Network Security*. Prentice Hall.

W.Diffie and M.Hellman (1976). New Directions in Cryptography. In *IEEE Transactions on Information Theory (22)*, pages 644–654. IEEE.

Wolfram, S. (1983). Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, 55(3):601–644.