The generalization of public key cryptosystem FAPKC4

TAO Renji and CHEN Shihua

Laboratory for Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100080, *China*

Abstract FAPKC4, a public key cryptosystem based on automata theory, is generalized so that component automata of compound automata in user's public key would not be restricted to memory finite automata. The generalized FAPKC4 can be used in encryption and implementing digital signatures as well.

Keywords: **public key cryptosystem, finite automata, invertibility.**

DIFFIE and Hellman[1] introduced the concept of public key cryptosystems. Many concrete schemes have been proposed and have witnessed important applications in the area of information security. Among the others, there are several public key cryptosystems based on finite automata theory, such as FAPKC0[2], FAPKC1 and FAPKC2[3], FAPKC93[1)], FAPKC3[4], and FAPKC4[5]. A user's public key of the above FAPKCs consists of a compound of two memory finite automata with some invertibility and initial states. On the other hand, in ref. [6], we research into the invertibility theory for general finite automata of which states consist of finite input history, finite output history and finite "inner state" history. Those theories laid a foundation for generalizing FAPKC so that component automata of compound automata in user's public key are such general finite automata. In ref. [7], we have made such generalization for FAPKC3. In this note, we generalize FAPKC4.

1 Theoretical foundation

Let $M = \langle X, Y, S^{p+1} \times X^r, \delta, \lambda \rangle$ be a finite automaton defined by

$$\begin{aligned} &\lambda(\langle s(i), \cdots, s(i-p), x(i-1), \cdots, x(i-r)\rangle, x(i)) = y(i), \\ &\delta(\langle s(i), \cdots, s(i-p), x(i-1), \cdots, x(i-r)\rangle, x(i)) \\ &= \langle s(i+1), \cdots, s(i+1-p), x(i), \cdots, x(i+1-r)\rangle, \end{aligned} \tag{1}$$

where

$$\begin{aligned} y(i) &= f(s(i), \cdots, s(i-p), x(i), \cdots, x(i-r)), \\ s(i+1) &= g(s(i), \cdots, s(i-p), x(i), \cdots, x(i-r)). \end{aligned} \tag{2}$$

Let $M^* = \langle Y, X, X^r \times S^{p+1} \times Y^\tau, \delta^*, \lambda^* \rangle$ be a finite automaton defined by

$$\begin{aligned} &\lambda^*(\langle x(i-1), \cdots, x(i-r), s(i), \cdots, s(i-p), y(i-1), \cdots, y(i-\tau)\rangle, y(i)) = x(i), \\ &\delta^*(\langle x(i-1), \cdots, x(i-r), s(i), \cdots, s(i-p), y(i-1), \cdots, y(i-\tau)\rangle, y(i)) \\ &= \langle x(i), \cdots, x(i+1-r), s(i+1), \cdots, s(i+1-p), y(i), \cdots, y(i+1-\tau)\rangle, \end{aligned} \tag{3}$$

where

$$\begin{aligned} x(i) &= f_\tau^*(x(i-1), \cdots, x(i-r), s(i), \cdots, s(i-p), y(i), \cdots, y(i-\tau)), \\ s(i+1) &= g(s(i), \cdots, s(i-p), x(i), \cdots, x(i-r)). \end{aligned} \tag{4}$$

Let $M' = \langle Z, Y, Y^k \times W^{n+1} \times Z^h, \delta', \lambda' \rangle$ be a finite automaton defined by

$$\begin{aligned} &\lambda'(\langle y(i-1), \cdots, y(i-k), w(i), \cdots, w(i-n), z(i-1), \cdots, z(i-h)\rangle, z(i)) = y(i), \\ &\delta'(\langle y(i-1), \cdots, y(i-k), w(i), \cdots, w(i-n), z(i-1), \cdots, z(i-h)\rangle, z(i)) \\ &= \langle y(i), \cdots, y(i+1-k), w(i+1), \cdots, w(i+1-n), z(i), \cdots, z(i+1-h)\rangle, \end{aligned} \tag{5}$$

where

$$\begin{aligned} y(i) &= \varphi(y(i-1), \cdots, y(i-k), w(i), \cdots, w(i-n), z(i), \cdots, z(i-h)), \\ w(i+1) &= \psi(y(i-1), \cdots, y(i-k), w(i), \cdots, w(i-n), z(i), \cdots, z(i-h)). \end{aligned} \tag{6}$$

Let $M'^* = \langle Y, Z, Z^h \times W^{n+1} \times Y^{\tau'+k}, \delta'^*, \lambda'^* \rangle$ be a finite automaton defined by

1) Gao Xiang, Finite automaton public key cryptosystems and digital signatures——analysis, design and implementation, *Dissertation* (in Chinese), Institute of Software, Chinese Academy of Sciences, Beijing, 1994.

$$\lambda'^{*}(\langle z(i-1),\cdots,z(i-h),\ w(i),\cdots,w(i-n),\ y(i-1),\ \cdots,y(i-\tau'-k)\rangle,\ y(i)) = z(i),$$
$$\delta'^{*}(\langle z(i-1),\cdots,z(i-h),\ w(i),\cdots,w(i-n),\ y(i-1),\ \cdots,y(i-\tau'-k)\rangle,\ y(i)) = \langle z(i),\cdots,z(i+1-h),\ w(i+1),\ \cdots,w(i+1-n),\ y(i),\cdots,y(i+1-\tau'-k)\rangle, \tag{7}$$

where

$$z(i) = \varphi^{*}_{\tau'}(z(i-1),\cdots,z(i-h),\ w(i),\cdots,w(i-n),\ y(i),\cdots,y(i-\tau'-k)),$$
$$w(i+1) = \psi(y(i-\tau'-1),\ \cdots,y(i-\tau'-k),\ w(i),\cdots,w(i-n),\ z(i),\cdots,z(i-h)). \tag{8}$$

Let $C'(M,M'^{*})$ be the compound finite automaton of M and M'^{*}, that is, $C'(M,M'^{*}) = \langle X, Z, Z^{h}\times W^{n+1}\times S^{\tau'+k+p+1}\times X^{\tau'+k+r},\ \delta'',\ \lambda''\rangle$, where

$$\lambda''(\langle z(i-1),\cdots,z(i-h),\ w(i),\ \cdots,w(i-n),\ s(i),\cdots,s(i-\tau'-k-p),\ x(i-1),\cdots,x(i-\tau'-k-r)\rangle,\ x(i))$$
$$= z(i)\delta''(\langle z(i-1),\cdots,z(i-h),\ w(i),\ \cdots,w(i-n),\ s(i),\cdots,s(i-\tau'-k-p),\ x(i-1),\cdots,x(i-\tau'-k-r)\rangle,\ x(i))$$
$$= \langle z(i),\cdots,z(i+1-h),\ w(i+1),\ \cdots,w(i+1-n),\ s(i+1),\cdots,s(i+1-\tau'-k-p),\ x(i),\cdots,x(i+1-\tau'-k-r)\rangle,$$

and

$$z(i) = \varphi'^{*}_{\tau'}(z(i-1),\cdots,z(i-h),\ w(i),\cdots,w(i-n),\ f(s(i),\cdots,s(i-p),\ x(i),\cdots,x(i-r)),\cdots,f(s(i-\tau'-k),\cdots,s(i-\tau'-k-p),\ x(i-\tau'-k),\cdots,x(i-\tau'-k-r))),$$
$$w(i+1) = \psi(f(s(i-\tau'-1),\cdots,s(i-\tau'-1-p),\ x(i-\tau'-1),\cdots,x(i-\tau'-1-r)),\ \cdots,f(s(i-\tau'-k),\cdots,s(i-\tau'-k-p),\ x(i-\tau'-k),\cdots,x(i-\tau'-k-r)),w(i),\cdots,w(i-n),\ z(i-1),\cdots,z(i-h)),$$
$$s(i+1) = g(s(i),\cdots,s(i-p),\ x(i),\cdots,x(i-r)). \tag{9}$$

We use $P_{\text{sig, inn}}(M^{*},\ M)$ to denote the following condition: for any state

$$s_0^{*} = \langle x(-1),\cdots,x(-r),s(0),\cdots,s(-p),y(-1),\cdots,y(-\tau)\rangle$$

of M^{*}, the state

$$s_0 = \langle s(0),\cdots,s(-p),x(-1),\cdots,x(-r)\rangle$$

of M matches s_0^{*} with delay τ.

We use $P_{\text{sig, out}}(M',\ M'^{*})$ to denote the following condition: for any state

$$s'_0 = \langle y(-1),\cdots,y(-k),\ w(0),\cdots,w(-n),\ z(-1),\cdots,z(-h)\rangle$$

of M' and any $z(0),\ z(1),\cdots\in Z$, if

$$y(0)y(1)\cdots = \lambda'(s'_0,\ z(0)z(1)\cdots),$$

then

$$z(0)z(1)\cdots = \lambda'^{*}(s'^{*}_{\tau'},\ y(\tau')y(\tau'+1)\cdots),$$

where

$$s'^{*}_{\tau'} = \langle z(-1),\cdots,z(-h),\ w(0),\cdots,w(-n),\ y(\tau'-1),\cdots,y(-k)\rangle.$$

Theorem 1. Assume that $P_{\text{sig,inn}}(M^{*},\ M)$ and $P_{\text{sig, out}}(M',\ M'^{*})$ hold. Given any state

$$s'_0 = \langle y(-1),\cdots,y(-k),\ w(0),\cdots,w(-n),z(-1),\cdots,z(-h)\rangle$$

of M' and any state

$$s^{*}_{-k} = \langle x(-k-1),\cdots,x(-k-r),\ s(-k),\cdots,s(-k-p),\ y(-k-1),\cdots,y(-k-\tau)\rangle$$

of M^{*}, let

$$x(i) = f^{*}_{\tau}(x(i-1),\cdots,x(i-r),\ s(i),\cdots,s(i-p),\ y(i),\cdots,y(i-\tau)),$$
$$s(i+1) = g(s(i),\cdots,s(i-p),\ x(i),\cdots,x(i-r)),$$
$$i = -k,\cdots,-1. \tag{10}$$

Denote

$$s_0^* = \langle x(-1),\cdots,x(-r),s(0),\cdots,s(-p),\ y(-1),\cdots,y(-\tau)\rangle.$$

For any $z(0),\ z(1),\ \cdots\in Z$, if

$$\lambda^*(s_0^*,\ \lambda'(s'_0,\ z(0)z(1)\cdots)) = x(0)x(1)\cdots \tag{11}$$

and

$$s'' = \langle z(-1),\cdots,z(-h),w(0),\cdots,w(-n),\ s(\tau+\tau'),\cdots,s(\tau-k-p),\\ x(\tau+\tau'-1),\cdots,x(\tau-k-r)\rangle,$$

then

$$\lambda''(s'',\ x(\tau+\tau')x(\tau+\tau'+1)\cdots) = z(0)z(1)\cdots, \tag{12}$$

where

$$s(i+1) = g(s(i),\cdots,s(i-p),x(i),\cdots,x(i-r)),\ i = 0,\cdots,\tau+\tau'-1. \tag{13}$$

Proof. Assume that (11) holds. Denote

$$\lambda'(s'_0,\ z(0)z(1)\cdots) = y(0)y(1)\cdots. \tag{14}$$

Then we have

$$\lambda^*(s_0^*, y(0)y(1)\cdots) = x(0)x(1)\cdots. \tag{15}$$

From (10),

$$\delta^*(s^*_{-k},\ y(-k)\cdots y(-1)) = s_0^*,$$
$$\lambda^*(s^*_{-k},\ y(-k)\cdots y(-1)) = x(-k)\cdots x(-1).$$

Using (15), it follows that

$$\lambda^*(s^*_{-k},\ y(-k)\cdots y(-1)y(0)y(1)\cdots) = x(-k)\cdots x(-1)x(0)x(1)\cdots. \tag{16}$$

From (10) and (13),

$$s(i+1) = g(s(i),\ \cdots,s(i-p),\ x(i),\cdots,x(i-r)),\\ i = -k,\ \cdots,\ -1,0,\cdots,\tau+\tau'-1. \tag{17}$$

Since $P_{\text{sig,inn}}(M^*,\ M)$ holds, from (16), there exist $y''(0),\cdots,y''(\tau-1)$ such that

$$\lambda(s_{-k},\ x(-k)\cdots,x(-1)x(0)x(1)\cdots)\\ = y''(0)\cdots y''(\tau-1)y(-k)\cdots y(-1)y(0)y(1)\cdots, \tag{18}$$

where $s_{-k} = \langle s(-k),\cdots,s(-k-p),\ x(-k-1),\cdots,x(-k-r)\rangle$. From the definition of M, using (17), it follows that

$$y(i-\tau) = f(s(i),\ \cdots,s(i-p),\ x(i),\cdots,x(i-r)),\\ i = \tau-k,\ \cdots,\tau+\tau'-1, \tag{19}$$

and

$$\delta(s_{-k},\ x(-k)\cdots x(\tau+\tau'-1)) = s_{\tau+\tau'}, \tag{20}$$

where $s_{\tau+\tau'} = \langle s(\tau+\tau'),\cdots,s(\tau+\tau'-p),\ x(\tau+\tau'-1),\cdots,x(\tau+\tau'-r)\rangle$. From (18) and (20), we have

$$\lambda(s_{\tau+\tau'},\ x(\tau+\tau')x(\tau+\tau'+1)\cdots) = y(\tau')y(\tau'+1)\cdots. \tag{21}$$

On the other hand, since $P_{\text{sig,out}}(M',\ M'^*)$ holds, from (14), we have

$$\lambda'^*(s'^*_{\tau'},\ y(\tau')y(\tau'+1)\cdots) = z(0)z(1)\cdots, \tag{22}$$

where $s'^*_{\tau'} = \langle z(-1),\cdots,z(-h),w(0),\cdots,w(-n),y(\tau'-1),\cdots,y(-k)\rangle$. Since (19) holds, from Theorem 1 in ref. [7], the state s'' of $C'(M,M')$ and the state $\langle s_{\tau+\tau'},\ s'^*_{-\tau'}\rangle$ of $C(M,\ M'^*)$ are equivalent. From (21) and (22), it immediately follows that

$$\lambda''(s'',x(\tau+\tau')x(\tau+\tau'+1)\cdots) = z(0)z(1)\cdots. \qquad \text{Q. E. D.}$$

We use $P_{\text{enc, inn}}(M,M^*)$ to denote the following condition: for any state

$$s_0 = \langle s(0),\cdots,s(-p),x(-1),\cdots,x(-r)\rangle$$

of M and any $x(0),\ x(1),\cdots\in X$, if

$$y(0)y(1)\cdots = \lambda(s_0,x(0)x(1)\cdots),$$

then

$$x(0)x(1)\cdots = \lambda^*(s_\tau^*,\ y(\tau)y(\tau+1)\cdots),$$

where

$$s_\tau^* = \langle x(-1),\cdots,x(-r),\ s(0),\cdots,s(-p),\ y(\tau-1),\cdots,y(0)\rangle.$$

We use $P_{\text{enc, out}}(M'^*, M')$ to denote the following condition: for any state

$$s_0'^* = \langle z(-1),\cdots,z(-h),w(0),\cdots,w(-n),\ y(-1),\cdots,y(-\tau'-k)\rangle$$

of M'^* and any $y(0),\ y(1),\cdots\in Y$, if

$$z(0)z(1)\cdots = \lambda'^*(s_0'^*, y(0)y(1)\cdots),$$

then

$$y(0)y(1)\cdots = \lambda'(s'_{\tau'},\ z(\tau')z(\tau'+1)\cdots),$$

where

$$s'_{\tau'} = \langle y(-1),\cdots,y(-k),\ w(\tau'),\cdots,w(\tau'-n),\ z(\tau'-1),\cdots,z(\tau'-h)\rangle,$$

and

$$w(i+1) = \psi(y(i-\tau'-1),\cdots,y(i-\tau'-k),\ w(i),\cdots,w(i-n),\ z(i),\cdots,z(i-h)),$$
$$i = 0,\cdots,\tau'-1. \tag{23}$$

Theorem 2. Assume that $P_{\text{enc, inn}}(M, M^*)$ and $P_{\text{enc, out}}(M'^*, M')$ hold. For any state

$$s_0'' = \langle z(-1),\cdots,z(-h),\ w(0),\cdots,w(-n),\ s(0),\cdots,s(-\tau'-k-p),\ x(-1),$$
$$\cdots,x(-\tau'-k-r)\rangle$$

of $C'(M,\ M'^*)$ and any $x(0),\ x(1),\cdots\in X$, if

$$\lambda''(s_0'', x(0)x(1)\cdots) = z(0)z(1)\cdots \tag{24}$$

and

$$y(0)y(1)\cdots = \lambda'(s'_{\tau'},\ z(\tau')z(\tau'+1)\cdots), \tag{25}$$

then we have

$$\lambda^*(s_\tau^*,\ y(\tau)\ y(\tau+1)\cdots) = x(0)x(1)\cdots, \tag{26}$$

where

$$s_\tau^* = \langle x(-1),\cdots,x(-r),\ s(0),\cdots,s(-p),\ y(\tau-1),\cdots,y(0)\rangle,$$
$$s'_{\tau'} = \langle y(-1),\cdots,y(-k),w(\tau'),\cdots,w(\tau'-n),\ z(\tau'-1),\cdots,z(\tau'-h)\rangle,$$

$w(1),\cdots,w(\tau')$ are computed by (23), and

$$y(i) = f(s(i),\cdots,s(i-p),\ x(i),\cdots,x(i-r)),\ i = -\tau'-k,\cdots,-1. \tag{27}$$

Proof. Denote

$$s_0 = \langle s(0),\cdots,s(-p),x(-1),\cdots,x(-r)\rangle,$$
$$s_0'^* = \langle z(-1),\cdots,z(-h),\ w(0),\cdots,w(-n),\ y(-1),\cdots,y(-\tau'-k)\rangle.$$

Since (27) holds, from Theorem 1 in ref. [7], the state s_0'' of $C'(M, M'^*)$ and the state $\langle s_0,\ s_0'^*\rangle$ of $C(M,\ M'^*)$ are equivalent. For any $x(0),\ x(1),\cdots\in X$, suppose that (24) and (25) hold. Denoting

$$\lambda(s_0,\ x(0)x(1)\cdots) = \bar{y}(0)\bar{y}(1)\cdots, \tag{28}$$

since s_0'' and $\langle s_0,\ s_0'^*\rangle$ are equivalent, from (24), we have

$$\lambda'^*(s_0'^*,\ \bar{y}(0)\bar{y}(1)\cdots) = z(0)z(1)\cdots. \tag{29}$$

Since $P_{\text{enc, out}}(M'^*,\ M')$ holds, from (29), we have

$$\bar{y}(0)\bar{y}(1)\cdots = \lambda'(s'_{\tau'},\ z(\tau')z(\tau'+1)\cdots).$$

From (25), it follows that

$$y(0)y(1)\cdots = \bar{y}(0)\bar{y}(1)\cdots.$$

Thus (28) yields

$$\lambda(s_0, x(0)x(1)\cdots) = y(0)y(1)\cdots. \tag{30}$$

Since $P_{\text{enc,inn}}(M, M^*)$ holds, from (30), we obtain (26). Q.E.D.

Corollary 1. In case of $n = -1$, replacing (27) by

$$y(i) = f(s(i),\cdots,s(i-p),\ x(i),\cdots,x(i-r)),\ i = -k,\cdots,-1,$$

the theorem still holds.

2 Basic algorithm

According to results in the previous section, the public key cryptosystem FAPKC4 may be extended. In this system, each user has a pair of keys in which one is public to all users and the other is secret.

Take alphabets X and Z as the column vector spaces over $GF(q)$ with dimensions l and m, respectively.

To construct a public key cryptosystem, choose a common q and l, and take $m = l$ for the sake of digital signature. In other words, all users use the same alphabet to communicate with each other.

Generating a public key and a secret key for a user, say A, is as follows.

(i) Construct a finite automaton $M = \langle X, Y, S^{p+1} \times X^r, \delta, \lambda \rangle$ defined by (1) and (2) and a finite automaton $M^* = \langle Y, X, X^r \times S^{p+1} \times Y^\tau, \delta^*, \lambda^* \rangle$ defined by (3) and (4), satisfying conditions $P_{\text{enc, inn}}(M, M^*)$ and $P_{\text{sig, inn}}(M^*, M)$.

(ii) Construct a finite automaton $M' = \langle Z, Y, Y^k \times W^{n+1} \times Z^h, \delta', \lambda' \rangle$ defined by (5) and (6) and a finite automaton $M'^* = \langle Y, Z, Z^h \times W^{n+1} \times Y^{\tau'+k}, \delta'^*, \lambda'^* \rangle$ defined by (7) and (8), satisfying conditions $P_{\text{enc, out}}(M'^*, M')$ and $P_{\text{sig, out}}(M', M'^*)$.

(iii) Construct the finite automaton $C'(M, M'^*) = \langle X, Z, Z^h \times W^{n+1} \times S^{\tau'+k+p+1} \times X^{\tau'+k+r}, \delta'', \lambda'' \rangle$ from M and M'^*.

(iv) Choose arbitrary state

$$s''_e = \langle z(-1), \cdots, z(-h), w(0), \cdots, w(-n), s(0), \cdots, s(-\tau'-k-p), x(-1), \cdots, x(-\tau'-k-r) \rangle$$

of $C'(M, M'^*)$.

Compute

$$y(i) = f(s(i), \cdots, s(i-p), x(i), \cdots, x(i-r)), \quad i = -\tau'-k, \cdots, -1.$$

Denote $s'_{\text{out},d} = \langle y(-1), \cdots, y(-\tau'-k) \rangle$.

Choose arbitrary state

$$s'_s = \langle \bar{y}(-1), \cdots, \bar{y}(-k), \bar{w}(0), \cdots, \bar{w}(-n), \bar{z}(-1), \cdots, \bar{z}(-h) \rangle$$

of M'. Choose arbitrary state

$$s^*_{-k} = \langle \bar{x}(-k-1), \cdots, \bar{x}(-k-r), \bar{s}(-k), \cdots, \bar{s}(-k-p), \bar{y}(-k-1), \cdots, \bar{y}(-k-\tau) \rangle$$

of M^* and compute

$$\begin{aligned} \bar{x}(i) &= f^*_\tau(\bar{x}(i-1), \cdots, \bar{x}(i-r), \bar{s}(i), \cdots, \bar{s}(i-p), \bar{y}(i), \cdots, \bar{y}(i-\tau)), \\ \bar{s}(i+1) &= g(\bar{s}(i), \cdots, \bar{s}(i-p), \bar{x}(i), \cdots, \bar{x}(i-r)), \\ i &= -k, \cdots, -1. \end{aligned}$$

Denote

$$s^*_s = \langle \bar{x}(-1), \cdots, \bar{x}(-r), \bar{s}(0), \cdots, \bar{s}(-p), \bar{y}(-1), \cdots, \bar{y}(-\tau) \rangle.$$

Denote

$$\begin{aligned} s''_{\text{out},v} &= \langle z(-1), \cdots, z(-h) \rangle, \\ s''_{\text{med},v} &= \langle w(0), \cdots, w(-n); \bar{s}(0), \cdots, \bar{s}(-\max(p, p+k-\tau)) \rangle, \\ s''_{\text{in},v} &= \langle \bar{x}(-1), \cdots, \bar{x}(-\max(r, r+k-\tau)) \rangle. \end{aligned}$$

(v) The public key of the user A is

$$C'(M, M'^*), \ s''_e, \ s''_{\text{out},v}, \ s''_{\text{med},v}, \ s''_{\text{in},v}, \ \tau + \tau'.$$

The secret key of the user A is

$$M', \ M^*, \ s'^*_{\text{out},d}, \ s'_s, \ s^*_s, \ \tau, \ \tau'.$$

Encryption. Any user, say B, wants to send to the user A a plaintext $x(0)x(1)\cdots x(b)$ in secret. B first suffixes any $\tau + \tau'$ digits, say $x(b+1)\cdots x(b+\tau+\tau')$, to the plaintext. Then using A's public key B computes the ciphertext $z(0)\cdots z(b+\tau+\tau')$ as follows:

$$z(0)z(1)\cdots z(b+\tau+\tau') = \lambda''(s''_e, x(0)x(1)\cdots x(b+\tau+\tau')).$$

Decryption. From the ciphertext $z(0)\cdots z(b+\tau+\tau')$, A can retrieve the plaintext using his (her) secret key as follows. First using M', $s'_{\text{out},d}$ in his (her) secret key and s''_e in his (her) public key, A

computes

$$y(0)y(1)\cdots y(b+\tau)=\lambda'(s'_{\tau'},\ z(\tau')z(\tau'+1)\cdots z(b+\tau+\tau')),$$

where

$$s'_{\tau'}=\langle y(-1),\cdots,y(-k),\ w(\tau'),\cdots,w(\tau'-n),\ z(\tau'-1),\cdots,z(\tau'-h)\rangle,$$

$w(1),\cdots,w(\tau')$ are computed by

$$w(i+1)=\psi(y(i-\tau'-1),\cdots,y(i-\tau'-k),\ w(i),\cdots,w(i-n),\ z(i),\cdots,z(i-h)),$$
$$i=0,\cdots,\tau'-1.$$

Then using M^* in his (her) secret key and s''_e in his (her) public key, from Theorem 2, A retrieves the plaintext:

$$x(0)x(1)\cdots x(b)=\lambda^*(s^*_\tau,\ y(\tau)y(\tau+1)\cdots y(b+\tau)),$$

where

$$s^*_\tau=\langle x(-1),\cdots,x(-r),s(0),\cdots,s(-p),\ y(\tau-1),\cdots,y(0)\rangle.$$

Signing. The user A to sign a message $\bar{z}(0)\cdots\bar{z}(b)$, first suffixes any $\tau+\tau'$ digits, say $\bar{z}(b+1),\cdots,\bar{z}(b+\tau+\tau')$, to the message. Then using his (her) secret key M', M^*, s'_s, s^*_s, computes the signature $\bar{x}(0)\bar{x}(1)\cdots\bar{x}(b+\tau+\tau')$ as follows:

$$\lambda^*(s^*_s,\ \lambda'(s'_s,\ \bar{z}(0)z(1)\cdots z(b+\tau+\tau')))=\bar{x}(0)x(1)\cdots\bar{x}(b+\tau+\tau').$$

Validation. Any user, say B, can verify the validity of the signature $\bar{x}(0)\bar{x}(1)\cdots\bar{x}(b+\tau+\tau')$ for $\bar{z}(0)\cdots\bar{z}(b)$ as follows. Using $C'(M,\ M'^*)$ and $s''_{\mathrm{out},v}$, $s''_{\mathrm{med},v}$, $s''_{\mathrm{in},v}$ in A's public key, B first computes

$$\bar{s}(i+1)=g(\bar{s}(i),\cdots,\bar{s}(i-p),\bar{x}(i),\cdots,\bar{x}(i-r)),\ i=0,\cdots,\tau+\tau'-1,$$

then computes

$$\lambda''(s''_v,\ \bar{x}(\tau+\tau')\bar{x}(\tau+\tau'+1)\cdots\bar{x}(b+\tau+\tau'))$$

which would coincide with the message $\bar{z}(0)\cdots\bar{z}(b)$ from Theorem 1, where

$$s''_v=\langle\bar{z}(-1),\cdots,\bar{z}(-h),\bar{w}(0),\cdots,\bar{w}(-n),\ s(\tau+\tau'),\cdots,\bar{s}(\tau-k-p),$$
$$\bar{x}(\tau+\tau'-1),\cdots,\bar{x}(\tau-k-r)\rangle.$$

Remark 1. Randomness of signing for special selection of parameters.

In case of $k=0$, s_0^* in Theorem 1 is arbitrarily given. It follows that $\bar{y}(-1),\cdots,\bar{y}(-\tau)$ in s^*_s in the signing process may be arbitrarily chosen. In this case, we may replace s^*_s in secret key by $s^*_{\mathrm{out},s}=\langle\bar{x}(-1),\cdots,\bar{x}(-r)\rangle$, $s^*_{\mathrm{med},s}=\langle\bar{s}(0),\cdots,s(-p)\rangle$.

In case of $k=0$ and $g(s(0),\cdots,s(-p),x(0),\cdots,x(-r))$ does not depend on $x(-r+\tau-1),\cdots,x(-r)$, (13) in Theorem 1 does not depend on $x(-r+\tau-1),\cdots,x(-r)$ which are arbitrarily given. It follows that $\bar{y}(-1),\cdots,y(-\tau)$ and $x(-r+\tau-1),\cdots,\bar{x}(-r)$ in s^*_s in the signing process may be arbitrarily chosen. In this case, we may replace s^*_s in secret key by $s^*_{\mathrm{out},s}=\langle\bar{x}(-1),\cdots,\bar{x}(-r+\tau)\rangle$, $s^*_{\mathrm{med},s}=\langle\bar{s}(0),\cdots,s(-p)\rangle$.

In case of $k=0$ and $g(s(0),\cdots,s(-p),x(0),\cdots,x(-r))$ does not depend on $s(-p+\tau-1),\cdots,s(-p)$, (13) in Theorem 1 does not depend on $s(-p+\tau-1),\cdots,s(-p)$ which are arbitrarily given. It follows that $\bar{y}(-1),\cdots,\bar{y}(-\tau)$ and $\bar{s}(-p+\tau-1),\cdots,\bar{s}(-p)$ in s^*_s in the signing process may be arbitrarily chosen. In this case, we may replace s^*_s in secret key by $s^*_{\mathrm{out},s}=\langle\bar{x}(-1),\cdots,\bar{x}(-r)\rangle$, $s^*_{\mathrm{med},s}=\langle\bar{s}(0),\cdots,\bar{s}(-p+\tau)\rangle$.

In case of $k=0$ and $g(s(0),\cdots,s(-p),x(0),\cdots,x(-r))$ does not depend on $x(-r+\tau-1),\cdots,x(-r)$ and $s(-p+\tau-1),\cdots,s(-p)$, (13) in Theorem 1 does not depend on $x(-r+\tau-1),\cdots,x(-r)$, $s(-p+\tau-1),\cdots,s(-p)$ which are arbitrarily given. It follows that $\bar{y}(-1),\cdots,\bar{y}(-\tau),\bar{x}(-r+\tau-1),\cdots,\bar{x}(-r)$ and $\bar{s}(-p+\tau-1),\cdots,\bar{s}(-p)$ in s^*_s in the signing process may be arbitrarily chosen. In this case, we may replace s^*_s in secret key by $s^*_{\mathrm{out},s}=\langle\bar{x}(-1),\cdots,\bar{x}(-r+\tau)\rangle$, $s^*_{\mathrm{med},s}=\langle\bar{s}(0),\cdots,s(-p+\tau)\rangle$.

Remark 2. Randomness of encryption for special selection of parameters.

(*To be continued on page 790*)

(*Continued from page* 789)

In case of $n = -1$, from Corollary 1 of Theorem 2, components $z(-\tau'-h-1),\cdots,z(-h)$, $s(-k-p-1),\cdots,s(-\tau'-k-p),x(-k-r-1),\cdots,x(-\tau'-k-r)$ of s''_0 are arbitrarily given. It follows that components $z(-\tau'-h-1),\cdots,z(-h),s(-k-p-1),\cdots,s(-\tau'-k-p)$, $x(-k-r-1),\cdots,x(-\tau'-k-r)$ of s''_0 in the encryption process may be arbitrarily chosen. In this case, we may replace s''_e in public key by $s''_{\text{out},e}=\langle z(-1),\cdots,z(-h+\tau')\rangle$, $s''_{\text{med},e}=\langle s(-1),\cdots,s(-k-p)\rangle$. $s''_{\text{in},e}=\langle x(-1),\cdots,x(-k-r)\rangle$.

Acknowledgement This work was supported by the National Natural Science Foundation of China (Grant No. 69773022).

References

1 Diffie, W., Hellman, M., New directions in cryptography, *IEEE Trans. on Information Theory*, 1976, IT-22: 644.
2 Tao Renji, Chen Shihua, A finite automaton public key cryptosystem and digital signatures, *Chinese J. of Computers* (in Chinese), 1985, 8: 401.
3 Tao Renji, Chen Shihua, Two varieties of finite automaton public key cryptosystem and digital signatures, *J. of Computer Science and Technology*, 1986, 1: 9.
4 Tao Renji, Chen Shihua, Chen Xuemei, FAPKC3: a new finite automaton public key cryptosystem, *J. of Computer Science and Technology*, 1997, 12: 289.
5 Tao Renji, Chen Shihua, A variant of the public key cryptosystem FAPKC3, *J. of Network and Computer Applications*, 1997, 20: 283.
6 Tao Renji, Chen Shihua, Constructing finite automata with invertibility by transformation method, *Advances in Cryptology——CHINACRYPT'98* (eds. Liu Mulan, Gong Qimin), Beijing: Science Press, 1998, 61.
7 Tao Renji, Chen Shihua, A note on the public key cryptosystem FAPKC3, *Advances in Cryptology——CHINACRYPT'98* (eds. Liu Mulan, Gong Qimin), Beijing: Science Press, 1998, 69.

(*Received February* 10, 1998)