Functional Encryption for Regular Languages

Brent Waters

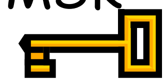
THE UNIVERSITY OF TEXAS AT AUSTIN

Public Key Encryption [DH76,M78,RSA78,GM84]

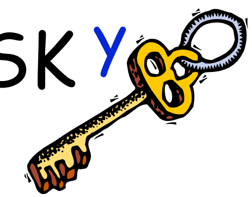
Avoid Prior Secret Exchange

Functional Encryption [SW05…]

Functionality: f(¢ , ¢)

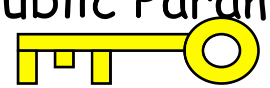
MSK


Authority

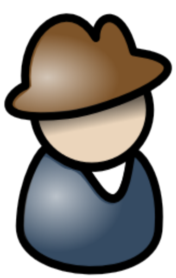
SK y


Key: y 2 {0,1}*

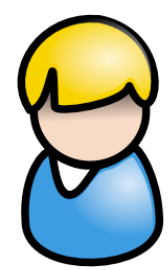
CT: x 2 {0,1}*

Public Params


Security: "Can only learn f(x,y)"




f(x,y)



"Key Policy" ABE [GPSW06]

Key: $y = \phi$ ⟵ Boolean Formula (or circuit)

CT: $x = (m, \vec{X} \in \{0,1\}^n)$ ⟵ Variables

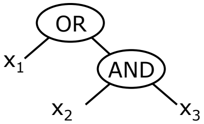
$$f(x = (m, \vec{X}), y) \rightarrow \begin{cases} m, \vec{X} \text{ if } \phi(\vec{X}) = \text{true} \\ \vec{X} \text{ if } \phi(\vec{X}) = \text{false} \end{cases}$$

"Public Index" ↗

Functionality: Evaluate formula, if true give message

Limitations

Key is a single formula/circuit

$\phi =$ 

Operates over fixed sized input

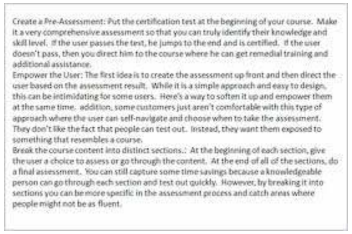
Fixed Size:

Form

Image

Arbitrary Length:

Text



Create a Pre-Assessment: Put the certification test at the beginning of your course. Make it a very comprehensive assessment so that you can truly identify their knowledge and skill level. If the user passes the test, he jumps to the end and is certified. If the user doesn't pass, then you direct him to the course where he can get remedial training and additional assistance.

Empower the User: The first idea is to create the assessment up front and then direct the user based on the assessment result. While it is a simple approach and easy to design, this can be intimidating for some users. Here's a way to soften it up and empower them at the same time. addition, some customers just aren't comfortable with this type of approach where the user can self-navigate and choose when to take the assessment. They don't like the fact that people can test out. Instead, they want them exposed to something that resembles a course.

Break the course content into distinct sections.: At the beginning of each section, give the user a choice to assess or go through the content. At the end of all of the sections, do a final assessment. You can still capture some time savings because a knowledgeable person can go through each section and test out quickly. However, by breaking it into sections you can be more specific in the assessment process and catch areas where people might not be as fluent.

Video

Goal: Functional Enc. for arbitrary length inputs

Regular Languages

Language is regular iff

strings accepted some Deterministic Finite Automata (DFA)

<u>Applications</u>

Search <[^>]*>

Firewall Rules (?i)^([^./]+\.)*(grooveshark\.com|gs-cdn\.net)(?![^/])

Determinstic Finite Automata (DFA)

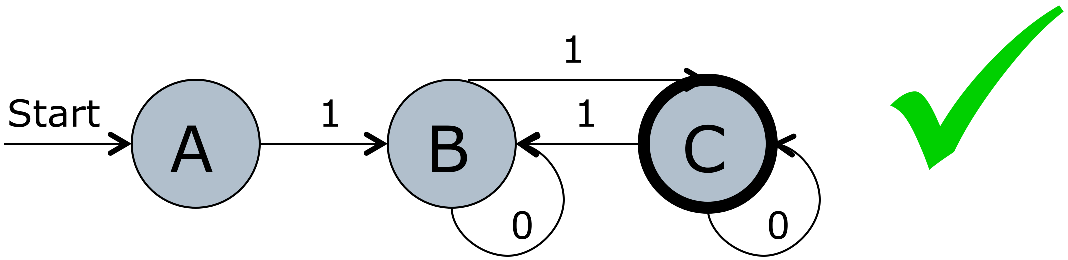
$$M = (Q, \Sigma, \delta, q_0, F)$$

Q	Set of states	$q_0 \in Q$	Start state
Σ	Alphabet	$F \in Q$	Accept states
$\delta : Q \times \Sigma \to Q$	Transition		

Note: Some Regular Expressions not efficiently expressible as DFAs.

A Simple Example

Language = "Begins with 1 and has even parity"



Accept(M,w)

w = 1 0 1 0

DFA-Based F.E. System

Key: $M = (Q, \Sigma, \delta, q_0, F)$ ← DFA

CT: $x = (m, w \in \Sigma^*)$ ← Arbitrary length string

$f(x = (m, \vec{X}), M) \rightarrow$ m, w if Accept(M, w)

w if Reject(M, w)

"Public Index" ↗

Functionality: Evaluate DFA M on w, if accepts give message

System Overview

Setting: Bilinear group G of order p

Key: $|Q|$ states, $D_0, \ldots, D_{|Q|-1} \stackrel{R}{\leftarrow} G$

CT: w: ℓ-symbol string, $s_0, \ldots, s_\ell \stackrel{R}{\leftarrow} \mathbb{Z}_p$

Decrypt: $e(g, D_x)^{s_j}$ ⟵ At state x after j symbols

<u>Three Mechanisms</u>

Initialization: Compute $e(g, D_0)^{s_0}$

Transition: $e(g, D_x)^{s_j} \rightarrow e(g, D_y)^{s_{j+1}}$ if $\delta(x, w_j) = y$

Completion: Recover message using $e(g, D_x)^{s_\ell}$ if $q_x \in F$

Setup

Input: Σ

1) Choose Bilinear group G of order p

2) $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p \quad g, z, h_{\mathsf{start}}, h_{\mathsf{end}}, \forall_{\sigma \in \Sigma}\ h_\sigma \stackrel{R}{\leftarrow} G$

Public Parameters: $e(g,g)^\alpha, g, z, h_{\mathsf{start}}, h_{\mathsf{end}}, \forall_{\sigma \in \Sigma}\ h_\sigma$

Master Secret: g^α

Encryption

Input: Message m, w: ℓ-symbol string

$$s_0, \ldots, s_\ell \xleftarrow{R} \mathbb{Z}_p$$

For $i = 1$ to ℓ $\quad C_{i,1} = g^{s_i}, \quad C_{i,2} = (h_{w_i})^{s_i} z^{s_{i-1}}$

"Linking"

Note: Only showing components for transition mechanism!

Key Generation

Input: $M = (Q, \delta, q_0, F)$

Define $(x, y, \sigma) \in \mathcal{T}$ if $\delta(x, \sigma) = y$

$$D_0, \ldots, D_{|Q|-1} \xleftarrow{R} G \qquad \forall t \in \mathcal{T} \;\; r_t \xleftarrow{R} \mathbb{Z}_p$$

$$\forall t = (x, y, \sigma) \in \mathcal{T}$$

$$K_{t,1} = D_x^{-1} z^{r_t}, \;\; K_{t,2} = g^{r_t}, \;\; K_{t,3} = D_y (h_\sigma)^{r_t}$$

Note: Only showing components for transition mechanism!

Transition Mechanism (of decryption)

Suppose $t = (x, y, \sigma) \in \mathcal{T}$ and $w_i = \sigma$

Compute:

$$e(C_{i-1,1}, K_{t,1}) e(C_{i,2}, K_{t,2})^{-1} e(C_{i,1}, K_{t,3})$$

$$= e(g^{s_{i-1}}, D_x^{-1} z^{r_t}) e((h_{w_i})^{s_i} z^{s_{i-1}}, g^{r_t})^{-1} e(g^{s_i}, D_y (h_\sigma)^{r_t})$$

$$= e(g, D_y)^{s_i} / e(g, D_x)^{s_{i-1}}$$

Transition: $e(g, D_x)^{s_{i-1}}$ to $e(g, D_y)^{s_i}$

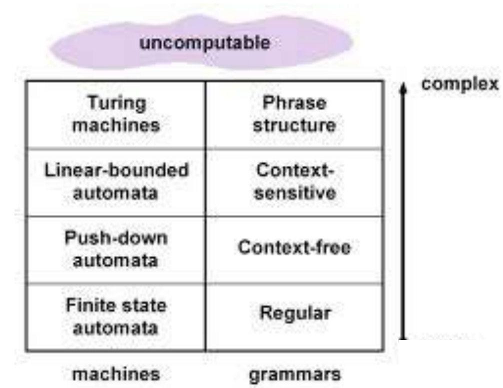
Summary & Three Problems

Functional Enc. for arbitrary length inputs: Achieved DFAs

Problems

(1) Support Non-deterministic Finite Automata (NFA)

(2) Climb the Chomsky Hierarchy



(3) Move past public index model

Thank you