Gate Elimination for Linear Functions and new Feebly Secure Constructions

Alex Davydow[1] Sergey I. Nikolenko[2]

[1]St. Petersburg Academic University, ul. Khlopina, 8, korp. 3, St. Petersburg, Russia, *adavydow@yandex.ru*

[2]Steklov Mathematical Institute, nab. r. Fontanka, 27, St. Petersburg, Russia, *sergey@logic.pdmi.ras.ru*

CSR 2011

Introduction

- Our subject is public key cryptosystems.
- No cryptosystem with public key has been proven to be secure.
- If a secure public key cryptosystem exists then $P \neq NP$.
- Moreover, asymptotic cryptography is kind of useless in practice: you would be interested in specific key sizes.
- To prove anything about specific key sizes, we have to talk about *circuit complexity*.

Introduction

- Of course, there are no nonlinear lower bounds in circuit complexity.
- But we can prove that *feebly secure* cryptosystems exist. Nikolenko and Hirsch constructed trapdoor functions which are $\frac{25}{22}$ times harder to break then to use.
- In this paper we will show an improvement of their construction allowing us to build a protocol which is $\frac{5}{4}$ harder to break then to use.
- From now on when speaking about complexity we will mean general circuit complexity.

Definitions

Fix functions $\mathrm{pi}, \mathrm{ti}, m, c : \mathbb{N} \to \mathbb{N}$. A *feebly trapdoor candidate* is a sequence of triples of circuits $\mathcal{C} = \{(\mathrm{Key}_n, \mathrm{Eval}_n, \mathrm{Inv}_n)\}_{n=1}^{\infty}$ where:

- $\{\mathrm{Key}_n\}_{n=1}^{\infty}$ is a family of sampling circuits $\mathrm{Key}_n : \mathbb{B}^n \to \mathbb{B}^{\mathrm{pi}(n)} \times \mathbb{B}^{\mathrm{ti}(n)}$,
- $\{\mathrm{Eval}_n\}_{n=1}^{\infty}$ is a family of evaluation circuits $\mathrm{Eval}_n : \mathbb{B}^{\mathrm{pi}(n)} \times \mathbb{B}^{m(n)} \to \mathbb{B}^{c(n)}$, and
- $\{\mathrm{Inv}_n\}_{n=1}^{\infty}$ is a family of inversion circuits $\mathrm{Inv}_n : \mathbb{B}^{\mathrm{ti}(n)} \times \mathbb{B}^{c(n)} \to \mathbb{B}^{m(n)}$

such that for every security parameter n, every seed $s \in \mathbb{B}^n$, and every input $\mathrm{m} \in \mathbb{B}^{m(n)}$

$$\mathrm{Inv}_n(\mathrm{Key}_{n,2}(s), \mathrm{Eval}_n(\mathrm{Key}_{n,1}(s), \mathrm{m})) = \mathrm{m},$$

where $\mathrm{Key}_{n,1}(s)$ and $\mathrm{Key}_{n,2}(s)$ are the first $\mathrm{pi}(n)$ bits ("public information") and the last $\mathrm{ti}(n)$ bits ("trapdoor information") of $\mathrm{Key}_n(s)$, respectively.

Definitions

- A circuit N *breaks* a feebly trapdoor candidate $\mathcal{C} = \{\mathrm{Key}_n, \mathrm{Eval}_n, \mathrm{Inv}_n\}$ on seed length n with probability r if, for uniformly chosen seeds $s \in \mathbb{B}^n$ and inputs $m \in \mathbb{B}^{m(n)}$,

$$\Pr_{(s,m)\in U}\left[N(\mathrm{Key}_{n,1}(s), \mathrm{Eval}_n(\mathrm{Key}_{n,1}(s), m)) = m\right] > r.$$

- A feebly trapdoor candidate $\mathcal{C} = \{\mathrm{Key}_n, \mathrm{Eval}_n, \mathrm{Inv}_n\}$ has *order of security k* with level $\frac{3}{4}$ if for every sequence of circuits $\{N_n\}_{n=1}^{\infty}$ that break f on every input length n with probability $\frac{3}{4}$,

$$\lim\inf_{n\to\infty} \min\left\{\frac{C(N_n)}{C(\mathrm{Key}_n)}, \frac{C(N_n)}{C(\mathrm{Eval}_n)}, \frac{C(N_n)}{C(\mathrm{Inv}_n)}\right\} \geq k.$$

Definitions

- We will work with linear Boolean functions.
- It is convenient to represent linear functions as matrices.
- These functions are still interesting because the following theorem holds:

Nonconstructive Bounds to Linear Functions

1. For every n there exists a constant δ_n such that the circuit complexity of all linear functions $\phi : \{0,1\}^n \to \{0,1\}^n$ does not exceed $\delta_n \frac{n^2}{\log n}$, and $\lim_{n\to\infty} \delta_n = 1$.
2. For every $n \geq 3$, there exists a linear Boolean function $\phi : \{0,1\}^n \to \{0,1\}^n$ with circuit complexity greater than $\frac{n^2}{2 \log n}$.

Gate Elimination

- To build secure constructions we need a method to prove lower bounds on complexity.
- *Gate elimination* is virtually the only method we have to prove lower bounds.

Gate Elimination

- Consider a function f and a circuit of minimal size C that computes it.
- Now substitute some value c for some variable x thus obtaining a circuit for the function $f\,|_{x=c}$.
- The original circuit C can now be simplified, because the gates that had this variable as inputs become either unary or constant.

Gate Elimination for Linear Functions

Idea 1

Suppose that for n steps, there is at least one gate to eliminate. Then $C(f) \geq n$.

- Simple example: a function f that nontrivially depends on all n inputs has $C(f) \geq n - 1$.

Gate Elimination for Linear Functions

Gate Elimination 1

Suppose that $\mathcal{P} = \{P_n\}_{n=1}^{\infty}$ is a series of predicates defined on matrices over $\mathbb{F}_2$ with the following properties:

- if $P_1(A)$ holds then $C_{3/4}(A) \geq 1$;
- if $P_n(A)$ holds then $P_m(A)$ holds for every $1 \leq m \leq n$;
- if $P_n(A)$ holds then, for every index i, $P_{n-1}(A_{-i})$ holds.

Then, for every matrix A with $\geq n+1$ different columns, if $P_n(A)$ holds for some n then $C(A) \geq n$ and, moreover, $C_{3/4}(A) \geq n$.

Gate Elimination for Linear Functions (Generalized)

Idea 1 is not optimal because on each elimination step, we count only one gate as eliminated, while sometimes we actually get two or more.

Idea 2

Suppose that for n steps, there exists an input in the circuit with two outgoing edges, and, moreover, in m of these cases both of these edges go to a gate (rather than a gate and an output). Then $C(f) \geq n + m$.

Gate Elimination for Linear Functions (Generalized)

Gate Elimination 2

Suppose that predicates $\mathcal{R} = \{R_n\}_{n=1}^{\infty}$ and $\mathcal{Q} = \{Q_m\}_{m=1}^{\infty}$ defined on matrices over $\mathbb{F}_2$ have the following properties:

- if $R_1(A)$ holds then $C(A) \geq 1$;
- if $R_n(A)$ holds then $R_k(A)$ holds for every $1 \leq k \leq n$;
- if $R_n(A)$ holds then, for every i, $R_{n-1}(A_{-i})$ holds;
- if $Q_1(A)$ holds then $C(A) \geq 1$;
- if $Q_m(A)$ holds then $Q_k(A)$ holds for every $1 \leq k \leq n$;
- if $Q_m(A)$ holds then, for every i, $Q_{m-1}(A_{-i})$ holds;
- if $Q_m(A)$ holds and A_{-i} has more zero rows than A then $Q_m(A_{-i})$ holds.

Then, for every matrix A with $\geq n+1$ columns, all of whose columns are different, if $R_n(A)$ and $Q_m(A)$ hold for some $n \geq m$ then $C(A) \geq n+m$ and, moreover, $C_{3/4}(A) \geq n+m$.

Gate Elimination for Linear Functions (Generalized)

- However, we are actually interested in the *total* number of gates eliminated rather than specifically eliminating one gate and two gates exactly (exact quantities and orderings may be hard to find).
- We call a nonzero entry *unique* if it is the only nonzero entry in its row.

Gate Elimination for Linear Functions (Generalized)

Gate Elimination 3

Suppose that $\mathcal{P} = \{P_n\}_{n=1}^{\infty}$ is a series of predicates defined on matrices over $\mathbb{F}_2$ with the following properties:

- if $P_1(A)$ holds then $C(A) \geq 1$;
- if $P_n(A)$ holds then $P_m(A)$ holds for every $1 \leq m \leq n$;
- if $P_n(A)$ holds then, for every index i, if the i^{th} column has no unique entries then $P_{n-2}(A_{-i})$ holds, otherwise $P_{n-1}(A_{-i})$ holds.

Then, for every matrix A with $\geq n+1$ different columns, if $P_n(A)$ holds for some n then $C(A) \geq n$ and, moreover, $C_{3/4}(A) \geq n$.

Corollaries

Using Gate Elimination we can obtain several simple algorithms to estimate complexity of linear Boolean functions.

Algorithm 1

Let $t, u \geq 1$. Assume also that A is a matrix with all colums different and, every row of A has at least u nonzero entries, and after removing any t columns of A, the matrix still has at least one row containing at least two nonzero entries. Then $C(A) \geq u + t$ and, moreover, $C_{3/4}(A) \geq u + t$.

Algorithm 2

Let $t \geq u \geq 2$. Assume that A is a $u \times t$ matrix with different columns, and each column of A has at least two nonzero elements (ones). Then $C(A) \geq 2t - u$ and, moreover, $C_{3/4}(A) \geq 2t - u$.

Corollaries

- While the first algorithm was introduced in Hirsch and Nikolenko's paper, the second is a new result.
- It is very simple but has several interesting applications.
- For example, with this idea we can build a matrix with complexity $2n + \frac{n}{\log(n)} - 2log(n) - 1$. Example of such a matrix is provided by cyclic shifts of Hamming code check matrices.

Block Diagonal Matrices

Block Diagonal Matrix Complexity

Suppose that a linear function χ is given by a block diagonal matrix

$$\begin{pmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & A_k \end{pmatrix},$$

and every A_j satisfies the conditions of Generalized Gate Elimination method with predicates $\mathcal{P}^j = \{P_n^j\}_{n=1}^{\infty}$, and $P_{n_j}^j(A_j)$ hold for every j. Then $C(\chi) \geq \sum_{j=1}^{k} n_j$.

New Feebly Secure Construction

By U_n, we denote the upper triangular square $n \times n$ matrix with a bidiagonal inverse:

$$U_n = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}, \qquad U_n^{-1} = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix};$$

note that U_n^2 is an upper triangular matrix with zeros and ones chequered. In what follows, we often write matrices that consist of other matrices as blocks; e.g., $\begin{pmatrix} U_n & U_n \end{pmatrix}$ is an $n \times 2n$ matrix consisting of two upper triangular blocks.

1. $C_{3/4}(U_n) = n - 1$.
2. $C_{3/4}(U_n^2) = n - 2$.
3. $C_{3/4}(U_n^{-1}) = n - 1$.
4. $C_{3/4}(\begin{pmatrix} U_n & U_n \end{pmatrix}) = 2n - 1$.
5. $3n - 6 \leq C_{3/4}(\begin{pmatrix} U_n^2 & U_n \end{pmatrix}) \leq C(\begin{pmatrix} U_n^2 & U_n \end{pmatrix}) \leq 3n - 3$.
6. $3n - 4 \leq C_{3/4}(\begin{pmatrix} U_n & U_n^{-1} \end{pmatrix}) \leq C(\begin{pmatrix} U_n & U_n^{-1} \end{pmatrix}) \leq 3n - 2$.

New Feebly Secure Construction

- We assume that lengths of public information pi, trapdoor information ti, message m, and the cipher c are the same and equal n.
- We let $ti = U_n \cdot pi$, $c = \begin{pmatrix} U_n^{-1} & U_n \end{pmatrix} \cdot \begin{pmatrix} m \\ pi \end{pmatrix}$.
- An adversary would have to compute the matrix $\begin{pmatrix} U_n & U_n \end{pmatrix} \cdot \begin{pmatrix} c \\ ti \end{pmatrix} = \begin{pmatrix} U_n & U_n^2 \end{pmatrix} \cdot \begin{pmatrix} c \\ pi \end{pmatrix}$.

Problem

Inversion without the trapdoor is harder than inversion with trapdoor, but encryption is about the same complexity as inversion without trapdoor.

Solving the Problem

- To solve this problem we will use a feebly one-way linear function A (one of Hiltgen's hard function with order of security up to 2).
- Their complexity follows from Algorithm 1, so we can stack them up into a block matrix.
- New protocol:

$$\begin{aligned} \mathrm{Key}_n &= \begin{pmatrix} U_n & 0 \\ 0 & I_n \end{pmatrix} \cdot \begin{pmatrix} s & s \end{pmatrix} = \begin{pmatrix} t_i \\ p_i \end{pmatrix}, \\ \mathrm{Eval}_n &= \begin{pmatrix} U_n^{-1} & U_n & 0 \\ 0 & 0 & A \end{pmatrix} \cdot \begin{pmatrix} m_1 \\ pi \\ m_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \\ \mathrm{Inv}_n &= \begin{pmatrix} U_n & U_n & 0 \\ 0 & 0 & A^{-1} \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ ti \\ c_2 \end{pmatrix} = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix}. \end{aligned}$$

Final Results

- Complexities of new protocol:

$$\begin{aligned} C_{3/4}(\mathrm{Key}_n) &= n-1, \\ C_{3/4}(\mathrm{Eval}_n) &= 3n+\lambda n+o(n), \\ C_{3/4}(\mathrm{Inv}_n) &= 2n+(2-\epsilon)\lambda n+o(n), \\ C_{3/4}(\mathrm{Adv}_n) &= 3n+(2-\epsilon)\lambda n+o(n). \end{aligned}$$

- The order of security of this construction is now:

$$\lim_{n\to\infty}\left(\min\left(\frac{C_{3/4}(\mathrm{Adv}_n)}{C(\mathrm{Eval}_n)},\frac{C_{3/4}(\mathrm{Adv}_n)}{C(\mathrm{Inv}_n)},\frac{C_{3/4}(\mathrm{Adv}_n)}{C(\mathrm{Key}_n)}\right)\right)= \\ =\min\left(\frac{3+(2-\epsilon)\lambda}{3+\lambda},\frac{3+(2-\epsilon)\lambda}{2+(2-\epsilon)\lambda}\right).$$

This expression reaches maximum for $\lambda=\frac{1}{1-\epsilon}$, and this maximum is $\frac{5-4\epsilon}{4-\epsilon}$, which tends to $\frac{5}{4}$ as $\epsilon\to 0$.

Thank you!

Thank you for your attention!