

# Completion Attacks and Weak Keys of Oleshchuk's Public Key Cryptosystem

**Heiko Stamer**

University of Kassel  
Department of Mathematics/Computer Science  
Heinrich-Plett-Straße 40, 34132 Kassel, Germany

`stamer@theory.informatik.uni-kassel.de`

76F7 3011 329D 27DB 8D7C 3F97 4F58 4EB8 FB2B E14F

6th International Conference on Cryptology in India  
Bangalore, December 10-12, 2005

# Outline

**Introduction**

**String-Rewriting Systems**

**Oleshchuk's Public Key Cryptosystem**

**Cryptanalysis**

Completion Attack

Experimental Results

**Concluding Remarks**

# Introduction

- ▶ Security of almost all practical Public Key Cryptography relies on the **hardness of number theoretic assumptions** like  
DLOG, FACTORING, CDH, RSA, S-RSA, DDH, ...
- ▲ Drawback: **No proof of hardness is known!**
- ▶ Idea: Use known hard (e.g.  $\mathcal{NP}$ -hard) problems like  
SVP, CVP, KNAPSACK, MinRank, SD, ...
- ▲ Drawback: **Gap between average- and worst-case complexity!**
  - ▶ Inefficient and complex implementations, weak keys, ...

Vladimir A. Oleshchuk.

*On Public-Key Cryptosystem Based on Church-Rosser String-Rewriting Systems.*  
First Annual International Conference on Computing and Combinatorics, 1995.

- ▶ Based on the undecidability of the **word problem in semigroups**
- ▶ Very simple representation by string-rewriting systems

# String-Rewriting Systems

## Definition (String-Rewriting System)

A **string-rewriting system** (SRS)  $T$  on  $\Sigma$  is a subset of  $\Sigma^* \times \Sigma^*$ . Each pair  $(l, r) \in T$  is called **rewriting rule**. The system  $T$  is **length-reducing**, if  $|l| > |r|$  for all  $(l, r) \in T$ .

## Definition (Reduction Relation, Thue Congruence)

Let  $u, v \in \Sigma^*$  be two arbitrary words and let  $T$  be a SRS on  $\Sigma$ .

$u \rightarrow_T v$  if there exist a rewriting rule  $(l, r) \in T$  and words  $x, y \in \Sigma^*$  such that  $u = xly$  and  $v = xry$ .

$u \rightarrow_T^* v$  is the reflexive, transitive closure of  $\rightarrow_T$ .

$u \leftrightarrow_T v$  if  $u \rightarrow_T v$  or  $v \rightarrow_T u$  holds, i.e.  $\leftrightarrow_T = \rightarrow_T \cup \rightarrow_T^{-1}$ .

$u \leftrightarrow_T^* v$  is the reflexive, transitive closure of  $\leftrightarrow_T$ , also known as the **Thue congruence** generated by  $T$ .

## Definition (Congruence, Congruence Class)

Two strings  $u, v \in \Sigma^*$  are **congruent modulo  $T$** , if  $u \leftrightarrow_T^* v$  holds. The **congruence class** of a string  $w \in \Sigma^*$  (modulo  $T$ ) is the set

$$[w]_T = \{z \in \Sigma^* \mid w \leftrightarrow_T^* z\}.$$

This notation is easily expandable to sets  $A \subseteq \Sigma^*$ :

$$[A]_T = \{y \in \Sigma^* \mid \exists x \in A \text{ such that } x \leftrightarrow_T^* y\}$$

## Definition (Refinement Property, Equivalence of SRS)

Let  $T_1, T_2$  be two arbitrary SRS.

- ▶  $T_1$  **refines**  $T_2$ , if  $x \leftrightarrow_{T_1}^* y$  implies  $x \leftrightarrow_{T_2}^* y$  for all  $x, y \in \Sigma^*$ .
- ▶  $T_1$  **equivalent** to  $T_2$ , if they generate the same Thue congruence.

## Definition

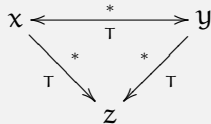
A word  $x \in \Sigma^*$  is **irreducible modulo  $T$** , if there exists no  $y \in \Sigma^*$  such that  $x \rightarrow_T y$ . The set  $\text{IRR}(T)$  denotes all **irreducible words**.

## Fact

If  $T$  is **finite**, then  $\text{IRR}(T)$  is regular and a DFA accepting this language can be constructed in polynomial time.

## Definition (Church-Rosser Property)

A SRS  $T$  is called **Church-Rosser**, if for all words  $x, y \in \Sigma^*$  with  $x \leftrightarrow_T^* y$  there exists a  $z \in \Sigma^*$  such that  $x \rightarrow_T^* z$  and  $y \rightarrow_T^* z$ .



## Problem (Uniform Word Problem)

**Instance:** A finite SRS  $T$  on  $\Sigma$  and two words  $x, y \in \Sigma^*$ .

**Question:** Are  $x, y$  congruent modulo  $T$ , i.e.  $x \leftrightarrow_T^* y$ ?

This problem is *undecidable* in general.

## Problem

**Instance:** A finite SRS  $T$  and a finite Church-Rosser SRS  $T'$ .

**Question:** Are  $T$  and  $T'$  equivalent, i.e.  $\leftrightarrow_T^* = \leftrightarrow_{T'}^*$ ?

This problem is *undecidable* in general. (O'Dunlaing, 1983)

## Problem

**Instance:** A finite *length-reducing* SRS  $T$ .

**Question:** Has  $T$  the Church-Rosser property?

This problem is *decidable* in time  $\mathcal{O}(|T|^3)$ . (Kapur et al., 1985)

## Fact (Book, 1982)

For a finite *length-reducing Church-Rosser SRS*  $T$ ,

1. each congruence class has a *unique irreducible element*, and
2. the word problem w.r.t.  $T$  is solvable in *linear time*.

## Coding Theory

A nonempty set  $C \subseteq \Sigma^*$  is called a *code*, if  $x_{i_1} \cdots x_{i_n} = x_{j_1} \cdots x_{j_m}$  implies  $x_{i_1} = x_{j_1}$  for all  $x_{i_1}, \dots, x_{i_n}, x_{j_1}, \dots, x_{j_m} \in C$ .

1. If  $C$  is a code, then any word  $x \in C^*$  has a *unique factorization* over  $C$ .
2. For a finite set  $C$  the code property is efficiently *decidable*.

# Oleshchuk's Public Key Cryptosystem (Setup)

W.l.o.g., let  $\Sigma = \{x_0, x_1\}$  be the plaintext alphabet and let  $\Delta$  be the ciphertext alphabet such that  $|\Delta| > |\Sigma|$ .

## Key Generation:

1. Choose a finite **length-reducing** Church-Rosser SRS  $T$  on  $\Delta$ .
2. Choose irreducible words  $u_1, \dots, u_t \in \text{IRR}(T)$  such that  $\{u_1, \dots, u_t\}^* \subseteq \text{IRR}(T)$  and  $\{u_1, \dots, u_t\}$  is a **code**.
3. Choose sets  $R_0, R_1 \subset \{u_1, \dots, u_t\}$  with  $R_0 \cap R_1 = \emptyset$ .
4. Choose **regular languages**  $L_i \subseteq [R_i]_T$  for  $i = 0, 1$ .
  - ▶ By construction we have  $L_0 \cap L_1 = \emptyset$ .
  - ▶ Representations for the  $L_i$ 's can be effectively constructed.
5. Choose a finite SRS  $S$  on  $\Delta$  that **refines**  $T$ , i.e. the congruence  $l \leftrightarrow_T^+ r$  holds for all  $(l, r) \in S$ .

Public key:  $(S, L_0, L_1)$

Secret key:  $(T, R_0, R_1)$

# Oleshchuk's Public Key Cryptosystem (Encryption)

The non-deterministic function  $\text{Encrypt} : \Sigma^* \rightarrow \Delta^*$  maps each single plaintext letter  $x_i \in \Sigma$  to a random word  $y \in [L_i]_S$ .

## Practical implementation:

1. Encode the plaintext  $m = x_{i_1} x_{i_2} \cdots x_{i_n}$  where  $x_{i_k} \in \Sigma$  by  $\hat{m} = \hat{x}_{i_1} \hat{x}_{i_2} \cdots \hat{x}_{i_n}$ , where each factor  $\hat{x}_{i_k} \in \Delta^*$  is randomly and uniformly chosen from the corresponding language  $L_{i_k}$ .
2. Rewrite  $\hat{m} \in L_{i_1} L_{i_2} \cdots L_{i_n}$  randomly and uniformly according to the rules of the public SRS  $S$ , i.e. construct a word  $c \in [L_{i_1} L_{i_2} \cdots L_{i_n}]_S$  such that  $\hat{m} \leftrightarrow_S^* c$  holds.

# Oleshchuk's Public Key Cryptosystem (Decryption)

Find an  $\hat{m} \in (L_0 \cup L_1)^*$  such that the congruence  $\hat{m} \leftrightarrow_S^* c$  holds.

In general, such a task is hard, because the SRS  $S$  may have an **intractable or even undecidable** word problem.

With knowledge of the secret key the decryption becomes easy:

1. Reduce  $c$  modulo  $T$  to obtain a unique normal form  $\tilde{m} \in \Delta^*$ .  
(linear time since  $T$  is Church-Rosser and length-reducing)
2. The factorization  $\tilde{m} = u_{i_1} u_{i_2} \cdots u_{i_n}$  with  $u_{i_k} \in R_{i_k}$  reveals the plaintext  $m = x_{i_1} x_{i_2} \cdots x_{i_n}$ .  
( $\{u_1, \dots, u_t\}$  is a code)

# Completion Attack (Basic Idea)

1. Use **completion procedures** (e.g. Knuth-Bendix algorithm) to construct a **Church-Rosser** SRS  $T'$  which is equivalent to  $S$ .

**Input:** The finite SRS  $S$  which is part of the public key.

**Output:** A finite length-reducing **Church-Rosser** SRS  $T'$  s.t.  $\leftrightarrow_S^* = \leftrightarrow_{T'}^*$ .

▲ Termination of the completion is undecidable in general.

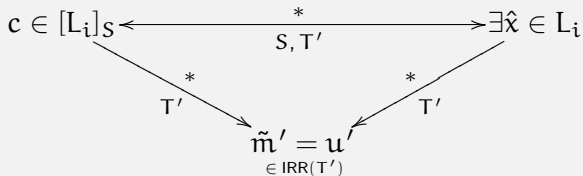
2. Reduce all words from  $L_i$  to their normal forms modulo  $T'$ , i.e. compute  $R'_i = \{u' \in \text{IRR}(T') \mid \exists \hat{x} \in L_i : \hat{x} \rightarrow_{T'}^* u'\}$ .
  - ▶ If  $L_i$  is finite, then  $R'_i$  can be efficiently computed.
3. Reduce the ciphertext  $c \in [L_i]_S$  modulo  $T'$  and obtain the unique normal form  $\tilde{m}'$ , i.e.  $c \rightarrow_{T'}^* \tilde{m}'$  holds.

**Idea:** Reduce the distinguishing problem  $i_1, i_2, \dots, i_n \in \{0, 1\}$  for a ciphertext  $c \in [L_{i_1} L_{i_2} \cdots L_{i_n}]_S$  to an easier question modulo  $T'$ .

### Lemma (One-bit Messages)

For  $i = 0, 1$  we have  $c \in [L_i]_S$ , if and only if  $\tilde{m}' \in R'_i$ .

### Proof.



Let  $\bar{L}'_0, \bar{L}'_1$  be finite “encoding languages” for a fixed ciphertext  $c \in [\bar{L}'_{i_1} \bar{L}'_{i_2} \cdots \bar{L}'_{i_n}]_S$ , and let  $\bar{R}'_0, \bar{R}'_1$  be the corresponding normal forms, i.e.  $\bar{R}'_i = \{u' \in \text{IRR}(T') \mid \exists \hat{x} \in \bar{L}'_i : \hat{x} \rightarrow_{T'}^* u'\}$ , for  $i = 0, 1$ .

**Fact:** The unique decryption property implies  $[\bar{L}'_0]_{T'} \cap [\bar{L}'_1]_{T'} = \emptyset$ .

### Lemma (Arbitrary Messages)

*Assuming that*

- a.  $([\bar{L}'_0]_{T'} \cup [\bar{L}'_1]_{T'})$  is a code, and
- b.  $([\bar{L}'_0]_{T'} \cup [\bar{L}'_1]_{T'})^* \subseteq \text{IRR}(T')$

*we have*

1.  $(\bar{R}'_0 \cup \bar{R}'_1)^* \subseteq \text{IRR}(T')$  and  $(\bar{R}'_0 \cup \bar{R}'_1)$  is a code, and hence
2.  $c \in [\bar{L}'_{i_1} \bar{L}'_{i_2} \cdots \bar{L}'_{i_n}]_S$  if and only if  $\tilde{m}' \in \bar{R}'_{i_1} \bar{R}'_{i_2} \cdots \bar{R}'_{i_n}$ .

# Experimental Results

*Proof of concept* implementation ( $\approx 1\,600$  lines of C++ code)

- ▶ Small instances ( $|\Delta| = 3$ ,  $|T| = 3$ ,  $6 \leq ||T|| \leq 12$ ), message (112 bit)
- ▶ KB-completion loop limited ( $\leq 3$  iterations and  $\leq 250$  critical pairs)
- ▶ 3 independent runs, 100 instances (randomly and uniformly chosen)

	$ S  = 3$	$ S  = 4$	$ S  = 5$
$ R_0 \cup R_1  =  L_0 \cup L_1  = 5$	23.3% vs. 15.3%	12% vs. 7.6%	4.3% vs. 2.3%
$ R_0 \cup R_1  =  L_0 \cup L_1  = 10$	19.6% vs. 11.3%	11% vs. 5%	4% vs. 1.6%
$ R_0 \cup R_1  =  L_0 \cup L_1  = 25$	23.6% vs. 11%	12% vs. 6%	4% vs. 1.3%

Table 1: Successful KB-completions versus successful ciphertext-only attacks

## Conclusions:

- 🏠 The attack works well, if the KB-completion is successful.
- ▲ Obviously, larger instances are much harder to attack.

# Concluding Remarks

## Countermeasures:

- ▶ Use infinite languages  $L_i$  (e.g. represented by left-linear grammars) to make the computation of  $R'_i$  more difficult
- ▶ Check whether  $S$  is easy to complete (undecidable in general)

## Further research:

- ▶ Use more sophisticated reduction orderings which may yield also efficiently decidable word problems
- ▶ Attack idea is applicable to other rewriting based cryptosystems, e.g. tree replacement (Samuel et al., 2002)

## Efficiency of the cryptosystem:

- ▶ The practical implementation raises a lot of questions:
  - ▶ “Cryptographically good” parameter setting
  - ▶ Efficient generation of  $S$  in a secure manner
  - ▶ Prevent a “brute-force” search in  $\leftrightarrow_S^*$  (exponential branching)
  - ▶ Extremely increased length of ciphertexts

# References



Vladimir A. Oleshchuk.

On Public-Key Cryptosystem Based on Church-Rosser String-Rewriting Systems.

Proceedings of COCOON '95, LNCS 959, 1995.



Vladimir A. Oleshchuk.

Church-Rosser Codes.

Proceedings of the 5th IMA Conference, LNCS 1025, 1996.



Ronald V. Book and Friedrich Otto.

String-Rewriting Systems.

Springer NY, ISBN 3-540979654, 1993.



S.C. Samuel, D.G. Thomas, P.J. Abisha, and K.G. Subramanian.

Tree Replacement and Public Key Cryptosystem.

Proceedings of INDOCRYPT '02, LNCS 2551, 2002.