Tatra Mt. Math. Publ. **33** (2006), 179–188

TATRA MOUNTAINS
Mathematical Publications

UNIVERSAL RE-ENCRYPTION OF SIGNATURES AND CONTROLLING ANONYMOUS INFORMATION FLOW

Marek Klonowski — Mirosław Kutyłowski — Anna Lauks — Filip Zagórski

ABSTRACT. Anonymous communication protocols, very essential for preserving privacy of the parties communicating, may lead to severe problems. A malicious server may use anonymous communication protocols for injecting unwelcome messages into the system so that their source can be hardly traced. Therefore anonymity and privacy protection on one side and protection against such phenomena as spam are so far contradictory goals.

We propose a mechanism that may be used to limit the mentioned side effects of privacy protection. During the proposed protocol each encrypted message admitted into the system is signed by a respective authority. Then, on its route through the network the encrypted message and the signature are re-encrypted universally. The purpose of universal re-encryption is to hide the routes of the messages from an observer monitoring the traffic.

Despite re-encryption, the signature of the authority remains valid. Depending on a particular application, verification of the signature is possible either off-line by anybody with access to the ciphertext and the signature or requires contacting the authority that has issued the signature.

Our work is an extension of recent works by Golle, Jakobsson, Juels and Syverson.

1. Introduction

There is an increasing demand on privacy of electronic communication. This is an obvious consequence of a growing role of electronic information flow. Sometimes electronic communication is used in very sensitive areas such as business negotiations. While encrypting a message sent from Alice to Bob protects its contents from an eavesdropper, it remains visible that Alice and Bob are communicating. Hence additional measures are necessary—protocols that provide

2000 Mathematics Subject Classification: 94A60, 11T71.
Keywords: anonymity, universal re-encryption, digital signature, undeniable signature.

This research has been partially supported in 2003–2004 by KBN, project 0 T00A 003 23.

anonymity for communicating parties. Many such protocols have been designed so far, however, the ultimate goal has not been reached yet.

One of the significant tools for constructing anonymous communication protocols is universal re-encryption [5]. In this scheme, given a ciphertext C with a decryption key K of an unknown user A, it is possible to recode C so that we obtain a new ciphertext C' of the same plaintext with decryption key K. An important point is that no knowledge of the public key used to create C is required.

Universal re-encryption may be used to build anonymous communication systems: encrypted messages are processed along the network and re-encrypted at each server. So one cannot see any connection between the incoming and the outcoming messages for a given server based on their shape. Such a solution can be applied in a network of mixes [4] or for onion-like communication, where the user determines the route of each message [6]. This is an excellent feature against an eavesdropper, however it does not protect against malicious servers that may insert own messages at the moment of re-encryption.

There are many attacks on the network of this type and even more might come in the future. This is an important issue, since not all servers can be trusted.

Admission protocol. We propose here a solution in which each message entering an anonymity system is checked before being admitted—either for its contents or for the origin (a message from a trusted party can be admitted automatically).

There is an authority maintaining the admission system: it runs a pool of special servers $\mathcal{A}$ devoted to admission procedures. They share the same private key used to issue a signature for each message processed through the anonymous communication system.

The general rules of the protocol are the following:

- Each message entering the system is encrypted with the public key of the recipient of the message.
- A message m entering the anonymous communication system must go through one of the servers from the pool $\mathcal{A}$. Such a server signs a ciphertext of m provided that either it knows m (and regards m as legal) or obtains m from a trusted user.
- Message m encoded as a ciphertext and accompanied by a signature of $\mathcal{A}$ is processed by the anonymous communication system. Each server on the route of m re-encrypts the ciphertext of m and transforms the signature.

- Signature's validity can be checked at any time, regardless of signature transformations performed. Neither the plaintext message m nor the destination place need to be known for verification.

Thus, for the purpose of the protocol we need to construct signatures that allow the transformations mentioned. We call them *URE signatures*. We consider here two scenarios: in the first case anybody can verify the signatures of $\mathcal{A}$. So it might be well suited for general systems like anonymous email systems. In the second scenario, one can verify a signature only with a help of the authority running the system. This is aimed for systems where one wish not to give up control over the system due, for instance, to law enforcement procedures. Note that the danger of corrupting one of these servers and gaining access to the private key is not really a critical issue, since a private key can be used by $\mathcal{A}$ for a limited period of time and can be updated at any moment.

Note that the signature scheme used by the servers in $\mathcal{A}$ and the encryption method must fit each other: if a ciphertext gets re-encrypted, we must be able to transform the signature so that it becomes a signature of the new ciphertext. It requires some special design of both schemes. Usual signature schemes guarantee that given a bit string u and a signature over u, one cannot derive a new signature over a bit string different from u. Here we are to break this principle: we shall derive signatures over bit strings $u' \neq u$, but only for cases when u and u' are ciphertexts of the same plaintext.

Previous work. A universal re-encryption scheme based on ElGamal encryption was introduced by Golle *et al.* [5]. The primary application of this technique was for networks of mixes [5, 4]. In [6] the same re-encryption technique was applied to redesign the onion encoding—it was used to escape from encapsulation principle used so far. This gave additional features of the protocol like automatic immunity against repetition attack without loosing the feature that an onion has to be processed by the servers along the onion path.

One can go one step further: Golle [4] introduced RSA-based signatures that can be re-encrypted. They were used to solve an important problem concerning mixes. Using this technique he constructs so-called reputable-mixes. A reputable mix can prove that every message it outputs corresponds to an input submitted by a user, without revealing which input it is. This mounts an effective measure against injecting "illegal outputs" by a mix (which is a common base of many attacks on mix systems).

Signatures allowing re-encryption. In order to achieve the mentioned goals we propose two signature schemes. The first one, called RSA-URE, is a simple extension of the scheme by Golle [4]. We made a slight modification of Golle's design, but there is a significant change concerning anonymity: our scheme does not require the knowledge of a public key used to create the signed ciphertext.

The second scheme proposed is based on undeniable signatures of Chaum, van Antwerpen [1].

In both cases the transformation of signatures achieves the same untraceability goals as re-encryption: it is infeasible to find any connection between the signatures before and after transformation except that both are valid signatures of $\mathcal{A}$.

Example applications. Scalable anonymous re-mailers can be implemented by networks of mixes. So it is an appealing idea to use universal re-encryption by each mix to unlink the incoming and the outcoming messages. However we might be forced to allow the mix network to grow beyond our direct control and it may happen that some of the mixes get malicious. To cope with this problem, we may use our protocol. Then any new message injected by a malicious mix server can be recognized, since it is not signed by the authority. The protocol proposed can be used against spam—the system would refuse to process email messages that have not been subject to a priori control. Note that such a control can be easily mounted locally (a special server in a LAN may easily recognize an outcoming spam), while it might be extremely hard to recognize a spam message once it is far from the alleged point of origin. Last but not least, we might be unhappy about any kind of control over our messages done by external parties, while an internal control within a local organization should be acceptable.

Another application of our scheme is to use URE-signatures as a kind of confirmation that a given message was processed by a particular anonymizer.

Why do we need undeniable URE-signatures? There are some business motivations (a company running admission points would like to preserve control over verification process as well), but there are more serious reasons as well. In some countries there are regulations that enforce the providers to enable eavesdropping of certain users (say by a decision of a court).

Our scheme enforces that the messages are presented to verification points run by the authority. At any of these points one can filter out the messages going to a suspect. One has to have his private key (obtained by some other law enforcement measures) and perform trial decryptions of the whole traffic going through such a server. Of course, encryption method guarantees that decryption of messages directed to other persons yields garbage only.

The law enforcement scheme proposed has also a nice feature that eavesdropping has its price: trial decryption of all messages with all keys becomes hard. So there is a technological barrier on playing "Big Brother".

2. URE signatures

Universal re-encryption. Let us recall the universal re-encryption scheme from [5]. We use a standard setting of the ElGamal encryption: p is such a prime number that the discrete logarithm is hard in the group $\mathbb{Z}_p^*$, g is a generator of $\mathbb{Z}_p^*$, the private key is a random $x < p-1$, and $y = g^x \bmod p$ is the corresponding public key. For the sake of simplicity from this point we skip the notation "mod p" when operations and equalities within $\mathbb{Z}_p^*$ are considered.

To encrypt a message m according to the ElGamal scheme we choose a number $k < p-1$ at random and compute $\alpha := g^k$, $\beta := m \cdot y^k$. Then (α, β) is a ciphertext of m. Universal re-encryption scheme is an extension of ElGamal scheme.

Encryption: To encrypt a message m for Alice, Bob generates values k_0 and k_1 uniformly at random. Then, the ciphertext of m is a quadruple:

$$(\alpha_0, \beta_0; \alpha_1, \beta_1) = \left(m \cdot y^{k_0}, g^{k_0}; y^{k_1}, g^{k_1}\right).$$

Decryption: Alice computes $m_0 = \alpha_0/\beta_0^x$, $m_1 = \alpha_1/\beta_1^x$, and accepts a message $m = m_0$, if and only if $m_1 = 1$.

Re-encryption: First random values k_0' and k_1' are chosen uniformly at random. Then the following tuple is computed and output as the re-encrypted ciphertext:

$$\left(\alpha_0 \cdot \alpha_1^{k_0'}, \beta_0 \cdot \beta_1^{k_0'}; \alpha_1^{k_1'}, \beta_1^{k_1'}\right).$$

In fact, the same algorithm can be implemented when we replace $\mathbb{Z}_p^*$ with a composite group $\mathbb{Z}_N^*$, where $N = pq$ is an RSA modulus. The expressions for ElGamal encryption, decryption and universal re-encryption operations can be used without any modification. A discussion on security of the ElGamal over composite groups can be found in [3], [7].

2.1 RSA-URE signatures

In this section we introduce an RSA-URE signature, which is a simple extension of the construction presented by Golle in [4].

Parameters: Let $N = pq$ be an RSA number, and let g be an arbitrary generator of $\mathbb{Z}_N^*$. Similarly as before, we skip the notation "mod N" whenever operations within $\mathbb{Z}_N$ are concerned.

Key setup: There are different key pairs in the system. The authority has a pair of keys that is used for signature creation and verification. Each user has a pair of keys that is used for protecting confidentiality of messages sent to this user.

The authority chooses e, which is co-prime with $\phi(N)$ and d such that $e \cdot d = 1 \bmod \phi(N)$. Then d is the private signing key, e is the public key for signature verification.

For generating encryption keys for a user we need some cooperation between the authority and the user. The parameter $\hat{g} := g^d$ is computed by the authority. In the meantime, the user chooses x uniformly at random and computes $y := g^x$. Finally, the authority computes $\hat{y} = y^d$ (so $\hat{y} = g^{dx} = \hat{g}^x$). Then the public encryption key is the tuple $(g, \hat{g}, y, \hat{y}, N)$ and the corresponding private key used for decryption is x.

Signature creation: For a message m, the authority creates the standard RSA signature m^d (at this moment we have to assume that some redundancy function has been applied and in fact m is the output of such a function for a payload message). Then the authority chooses values k_0, k_1 uniformly at random. Then it computes the following tuple, which is the final signed ciphertext:

$$(\alpha_0, \beta_0; \alpha_1, \beta_1; \alpha_2, \beta_2; \alpha_3, \beta_3) := \left(m \cdot y^{k_0}, g^{k_0}; y^{k_1}, g^{k_1}; m^d \cdot \hat{y}^{k_0}, \hat{g}^{k_0}; \hat{y}^{k_1}, \hat{g}^{k_1}\right).$$

Re-encryption: Values k_0', k_1' are chosen uniformly at random. Then the re-encrypted message is obtained through the following formula:

$$\left(\alpha_0 \cdot \alpha_1^{k_0'}, \beta_0 \cdot \beta_1^{k_0'}; \alpha_1^{k_1'}, \beta_1^{k_1'}; \alpha_2 \cdot \alpha_3^{k_0'}, \beta_2 \cdot \beta_3^{k_0'}; \alpha_3^{k_1'}, \beta_3^{k_1'}\right).$$

Note that we obtain a tuple of the following form

$$\begin{aligned}
&\Big(m \cdot y^{k_0} \cdot (y^{k_1})^{k_0'}, g^{k_0} \cdot (g^{k_1})^{k_0'};\ (y^{k_1})^{k_1'}, (g^{k_1})^{k_1'}; \\
&\qquad\qquad m^d \cdot \hat{y}^{k_0} \cdot (\hat{y}^{k_1})^{k_0'}, \hat{g}^{k_0} \cdot (\hat{g}^{k_1})^{k_0'};\ (\hat{y}^{k_1})^{k_1'}, (\hat{g}^{k_1})^{k_1'}\Big) \\
&\quad = \Big(m \cdot y^{k_0 + k_1 \cdot k_0'}, g^{k_0 + k_1 \cdot k_0'}; \\
&\qquad\qquad y^{k_1 \cdot k_1'}, g^{k_1 \cdot k_1'}; m^d \cdot \hat{y}^{k_0 + k_1 \cdot k_0'}, \hat{g}^{k_0 + k_1 \cdot k_0'};\ \hat{y}^{k_1 \cdot k_1'}, \hat{g}^{k_1 \cdot k_1'}\Big).
\end{aligned}$$

So it is just another RSA-URE signature, namely one constructed with random numbers $k_0 + k_1 \cdot k_0'$ and $k_1 \cdot k_1'$.

Signature verification: If an RSA-URE signature is correct, then for some k we have $\alpha_0 = m \cdot y^k$, $\alpha_2 = m^d \cdot \hat{y}^k$, so $\alpha_2 = \alpha_0^d$. Hence the verifier accepts the signature if and only if $\alpha_0 = \alpha_2^e$.

So far we have assumed that the authority has access to the message m. This corresponds to a scenario when the authority has to check m for its contents. In this case we simply let the authority construct both the signature and the ciphertext.

In the case when it is desirable to preserve confidentiality of m, we can use blinding. So assume that Alice is sending a message m to Bob. First Alice chooses k at random such that k and N are co-prime. Then Alice computes

$$t := \left(m \cdot y^{k_0} \cdot k^e, g^{k_0} \cdot k^e; y^{k_1} \cdot k^e, g^{k_1} \cdot k^e\right)$$

and sends t to the authority. In the next step the authority raises each component to the power d and obtains

$$t^d = \left(m^d \cdot y^{d\cdot k_0} \cdot k,\ g^{d\cdot k_0} \cdot k;\ y^{d\cdot k_1} \cdot k,\ g^{d\cdot k_1} \cdot k\right).$$

Then it sends the result to Alice. She divides it by the blinding factor k and obtains

$$\frac{t^d}{k} = \left(m^d \cdot y^{d\cdot k_0},\ g^{d\cdot k_0};\ y^{d\cdot k_1},\ g^{d\cdot k_1}\right).$$

Finally, Alice creates the standard RSA-URE signature:

$$(\alpha_0, \beta_0; \alpha_1, \beta_1; \alpha_2, \beta_2; \alpha_3, \beta_3) := \left(m\cdot y^{k_0}, g^{k_0}; y^{k_1}, g^{k_1}; m^d \cdot y^{d\cdot k_0}, g^{d\cdot k_0}; y^{d\cdot k_1}, g^{d\cdot k_1}\right),$$

which is ready to be sent to Bob.

3. Undeniable URE signatures

First let us recall undeniable signature scheme of Chaum, van Antwerpen [1]. Such a signature can be verified only in cooperation with the author of the signature. The author can prove invalidity of a fake signature, while such a proof is practically impossible for a legitimate signature.

Preliminary settings: Let p, q be primes, such that $p = 2q + 1$. Let G be a subgroup of $\mathbb{Z}_p^*$ of order q. We assume that the discrete logarithm problem for G is hard. Let g be a generator of G. From now on, we skip the notation "mod p" whenever the computations are performed in $\mathbb{Z}_p^*$.

Key setup: Alice chooses her private key $d \in \{1, \ldots, q-1\}$ at random, then the corresponding public key e is computed as $e := g^d$.

Signature creation: $s := m^d$ is Alice's signature of a message m.

Signature verification:

- The verifier chooses $i, j \in \{1, \ldots, q-1\}$ uniformly at random, computes $z := s^i \cdot e^j$ and presents z to Alice.
- Alice computes $w := (z)^{d^{-1} \bmod q}$ and presents w to the verifier.
- The verifier computes $w' := m^i \cdot g^j$ and accepts the signature s, if $w = w'$.

If $w \neq w'$, the same steps are repeated:

- The verifier chooses $\widehat{i}, \widehat{j} \in \{1, \ldots, q-1\}$ uniformly at random, computes $\widehat{z} := s^{\widehat{i}} \cdot e^{\widehat{j}}$ and transmits $\widehat{z}$ to Alice.
- Alice computes $\widehat{w} := (\widehat{z})^{d^{-1} \bmod q}$ and sends $\widehat{w}$ to the verifier.
- The verifier computes $\widehat{w'} := m^{\widehat{i}} \cdot g^{\widehat{j}}$. If $\widehat{w} = \widehat{w'}$, the verifier accepts the signature and ends up the protocol.

If $\widehat{w} \neq \widehat{w'}$, then:

- The verifier computes $c := (w \cdot g^{-j})^{\widehat{i}}$ and $\widehat{c} := (\widehat{w} \cdot g^{-\widehat{j}})^{i}$. If $c = \widehat{c}$, the signature was proved to be invalid. If $c \neq \widehat{c}$, the signature s is regarded as valid, but Alice tried to deny it.

3.1 Undeniable-URE signature

In this section we present an undeniable-URE signature, a scheme which is a combination of universal re-encryption and Chaum-van Antwerpen undeniable signatures. We use parameters p, q, g and G chosen as in the previous section.

Key setup: First, each participant establishes a pair of keys that are used to encode and decode messages sent to this participant. So Bob chooses his private key x at random and computes the corresponding public key $y := g^x$. Second, the authority chooses a private key d at random and computes the corresponding public key $e := g^d$.

Ciphertext and signature creation: Assume that Alice sends a message m to Bob. First consider the case when Alice is obliged to show m to the authority.

In this case the authority generates values k_0 and k_1 uniformly at random. Then it computes $s := m^d$. Finally, a signed ciphertext of m is created as a tuple $(\alpha_0, \beta_0; \alpha_1, \beta_1; \alpha_2, \beta_2; \alpha_3, \beta_3)$ of the form:

$$\left(m \cdot y^{k_0}, g^{k_0}; y^{k_1}, g^{k_1}; s \cdot y^{dk_0}, g^{dk_0}; y^{dk_1}, g^{dk_1}\right).$$

The case, when m has to be hidden from the authority, is similar. Now Alice chooses values k_0 and k_1, computes a quadruple

$$\left(m \cdot y^{k_0}, g^{k_0}; y^{k_1}, g^{k_1}\right)$$

and presents it to the authority. Then the authority computes the missing components by rising the numbers obtained to the power d.

Re-encryption: Values k_0' and k_1' are generated uniformly at random. A re-encrypted message is computed according to the following formula:

$$\left(\alpha_0 \cdot \alpha_1^{k_0'}, \beta_0 \cdot \beta_1^{k_0'}; \alpha_1^{k_1'}, \beta_1^{k_1'}; \alpha_2 \cdot \alpha_3^{k_0'}, \beta_2 \cdot \beta_3^{k_0'}; \alpha_3^{k_1'}, \beta_3^{k_1'}\right).$$

Signature verification: In cooperation with the authority, Eve can verify a message $(\alpha_0, \beta_0; \alpha_1, \beta_1; \alpha_2, \beta_2; \alpha_3, \beta_3)$ which allegedly has a form $(m \cdot y^k, g^k; y^l, g^l; m^d \cdot y^{d \cdot k}, g^{d \cdot k}; y^{d \cdot l}, g^{d \cdot l})$ for some k and l.

- Eve chooses $i, j \in \{1, \ldots, q-1\}$ uniformly at random, computes $z := \left(\alpha_2^i \cdot e^j, \beta_2^i \cdot e^j; \alpha_3^i \cdot e^j, \beta_3^i \cdot e^j\right)$ and presents z to the authority.
- The authority computes

$$w := (z)^{d^{-1}} = \left((\alpha_2^i \cdot e^j)^{d^{-1}}, (\beta_2^i \cdot e^j)^{d^{-1}}; (\alpha_3^i \cdot e^j)^{d^{-1}}, (\beta_3^i \cdot e^j)^{d^{-1}}\right)$$

and presents w to Eve.

- Eve computes: $w' := \left(\alpha_0^i \cdot g^j, \beta_0^i \cdot g^j; \alpha_1^i \cdot g^j, \beta_1^i \cdot g^j\right)$ and accepts the signature, if $w = w'$.

If the signature of the authority is correct, then:

$$\begin{aligned} w &= \left(\alpha_2^{id^{-1}} \cdot e^{jd^{-1}}, \beta_2^{id^{-1}} \cdot e^{jd^{-1}}; \alpha_3^{id^{-1}} \cdot e^{jd^{-1}}, \beta_3^{id^{-1}} \cdot e^{jd^{-1}}\right) \\ &= \left(m^{did^{-1}} \cdot y^{dkid^{-1}} \cdot g^{djd^{-1}}, g^{dkid^{-1}} \cdot g^{djd^{-1}}; y^{dlid^{-1}} \cdot g^{djd^{-1}}, g^{dlid^{-1}} \cdot g^{djd^{-1}}\right) \\ &= \left(m^i \cdot y^{ik} \cdot g^j, g^{ik+j}; y^{il} \cdot g^j, g^{il+j}\right). \end{aligned}$$

The tuple w' computed by Eve equals

$$w' = \left((m \cdot y^k)^i \cdot g^j, (g^k)^i \cdot g^j; (y^l)^i \cdot g^j, (g^l)^i \cdot g^j\right),$$

which is the same as w.

If $w \neq w'$, then:

- Eve chooses $\widehat{i}, \widehat{j} \in \{1, \ldots, q-1\}$ uniformly at random, computes $\widehat{z} := \left(\alpha_2^{\widehat{i}} \cdot e^{\widehat{j}}, \beta_2^{\widehat{i}} \cdot e^{\widehat{j}}; \alpha_3^{\widehat{i}} \cdot e^{\widehat{j}}, \beta_3^{\widehat{i}} \cdot e^{\widehat{j}}\right)$ and presents it to the authority.
- The authority computes

$$\widehat{w} := \left((\alpha_2^{\widehat{i}} \cdot e^{\widehat{j}})^{d^{-1}}, (\beta_2^{\widehat{i}} \cdot e^{\widehat{j}})^{d^{-1}}; (\alpha_3^{\widehat{i}} \cdot e^{\widehat{j}})^{d^{-1}}, (\beta_3^{\widehat{i}} \cdot e^{\widehat{j}})^{d^{-1}}\right)$$

  and presents $\widehat{w}$ to Eve.
- Eve computes: $\widehat{w'} = \left(\alpha_0^{\widehat{i}} \cdot g^{\widehat{j}}, \beta_0^{\widehat{i}} \cdot g^{\widehat{j}}; \alpha_1^{\widehat{i}} \cdot g^{\widehat{j}}, \beta_1^{\widehat{i}} \cdot g^{\widehat{j}}\right)$. If $\widehat{w} = \widehat{w'}$, she accepts the signature.

If $\widehat{w'} \neq \widehat{w}$, then:

- Eve computes: $c := \left(w \cdot g^{-j}\right)^{\widehat{i}}$ and $\widehat{c} = \left(\widehat{w} \cdot g^{-\widehat{j}}\right)^i$.
- If $c = \widehat{c}$, then she considers this signature invalid. If $c \neq \widehat{c}$, then Eve knows that the authority is cheating.

4. Conclusions and open problems

One can see from the current experience that constructing URE signatures is relatively straightforward for signature schemes that are based on exponentiations only. However, if exponentiations and arithmetic operations on the exponents are performed for issuing a signature (as in the case of DSA signatures), then there is no easy solution of this kind.

REFERENCES

[1] CHAUM, D.—VAN ANTWERPEN, H.: *Undeniable signatures*, in: Proceedings of CRYPTO '89, Lecture Notes in Comput. Sci., Vol. 435, Springer-Verlag, Berlin, 1990, pp. 212–216.

[2] FAIRBROTHER, P.: *An improved construction for universal re-encryption*, in: Proceedings of Privacy Enhancing Technologies (PET '04) Lecture Notes in Comput. Sci., Vol. 3424, Springer-Verlag, Berlin, 2005, p. 79.

[3] FRANKLIN, M.—HABER, S. S.: *Joint encryption and message-efficient secure computation*, J. Cryptology **9** (1996), 217–232.

[4] GOLLE, P.: *Reputable mix networks*, in: Proceedings of Privacy Enhancing Technologies (PET '04) Lecture Notes in Comput. Sci., Vol. 3424, Springer-Verlag, Berlin, 2005, p. 51.

[5] GOLLE, P.—JAKOBSSON, M.—JUELS, A.—SYVERSON, P.: *Universal re-encryption for mixnets*, in: Proceedings of RSA-CT 2004, Lecture Notes in Comput. Sci., Vol. 2964, Springer-Verlag, Berlin, 2004, pp. 163–178.

[6] GOMULKIEWICZ, M.—KLONOWSKI, M.—KUTYŁOWSKI, M.: *Onion routing based on universal re-encryption immune against repetitive attack*, in: Proceedings of Workshop on Information Security Applications (WISA '04), Lecture Notes in Comput. Sci., Vol. 3325, Springer-Verlag, Berlin, 2005, p. 400.

[7] MCCURLEY, M.: *A key distribution system equivalent to factoring*, J. Cryptology **1** (1988), 95–105.

Received October 3, 2004

Institute of Mathematics
and Computer Science
Wrocław University of Technology
Wybrzeż e Wyspiańskiego 27
PL–50-370 Wrocław
POLAND

E-mail: Marek.Klonowski@im.pwr.wroc.pl
Miroslaw.Kutylowski@pwr.wroc.pl
Filip.Zagorski@im.pwr.wroc.pl
Anna.Lauks@im.pwr.wroc.pl