# Efficient and Non-interactive Timed-Release Encryption

Julien Cathalo*, Benoît Libert**, and Jean-Jacques Quisquater

UCL Crypto Group,
Place du Levant, 3. B-1348 Louvain-La-Neuve, Belgium
{cathalo, libert, jjq}@dice.ucl.ac.be

**Abstract.** This paper revisits the important problem of sending a message "into the future" in such a way that no communication is needed between the server and other entities. This problem was recently re-investigated by Blake and Chan who showed a scalable non-interactive solution without considering a formal security model. We fill this gap by introducing a new stringent model tailored to the non-interactive setting. We then propose a new construction fitting our model and we show that it is more efficient than the recent non-interactive proposal (for which we also give a security proof in our model). We then explain how to provide our scheme and the one of Blake and Chan with an additional security property that strengthens the anonymity of receivers.

**Keywords:** timed-release encryption, formal models, provable security.

## 1 Introduction

The problem of sending a message "into the future", i.e. encrypting a message so that its recipient cannot decrypt it prior to a pre-determined instant chosen by the sender, has been found to have many real-world applications such as electronic auctions, key escrow, scheduled payment methods, sealed-bid auctions, lotteries, etc.. It was first suggested by May [26] in 1993 and further studied by Rivest, Shamir and Wagner [29].

Two essential approaches have been investigated to solve the problem: the time-lock puzzle approach ([6,29,25,14,22,23]) and the trusted server approach ([17,26,29,12]). In the former, the receiver of an encrypted message has to invest in a significant computational effort to solve a reasonably small-size problem before obtaining the message. This approach does not involve a server but it turns out to be computationally expensive for the receiver and only solves the problem with approximately controllable release-times depending on the receiver's computational power and on the moment at which the decryption operation is started. Sending a message that can be read at a precise moment (say 12:00am, July 31, 2005 GMT for example) turns out to be difficult using this approach.

* This author's work is supported by *Walloon Region / WIST-MAIS project.*
** This author thanks the DGTRE's *First Europe Program* of the *Walloon Region* and the *European Social Fund.*

S. Qing et al. (Eds.): ICICS 2005, LNCS 3783, pp. 291–303, 2005.

On the other hand, in the trusted server approach, a trusted entity providing a common and absolute time reference is necessary to synchronize senders and receivers. Ideally, the server should have as few interactions as possible with senders and receivers. Up to very recently, the latter requirement was not satisfied by server-based solutions. In a system proposed by May ([26]), the server is an escrow agent storing messages and releasing them at specified times. Another method used by Rivest et al. [29] also requires interactions between the server and senders who must reveal their identity and their message's release-time.

In 1999, Di Crescenzo et al. ([17]) proposed a protocol, supported by a formal security model, and wherein senders are anonymous and do not have to interact with the server. Unfortunately, the latter has to engage in a conditional oblivious transfer protocol with receivers. As a result, the latters are not anonymous and the protocol is subject to denial-of-service attacks.

In 2001, when Boneh and Franklin published their famous identity based encryption (IBE) scheme ([12]), they also mentioned encryption "for the future" as a possible application. Their idea was to use identities augmented with release times as public keys. This solution is not scalable for small time granularities as the trusted private key generator has to deliver new private keys to each user at the start of each time period. Other IBE-based approaches ([16,27,11]) consider release-times as identities and trusted authorities as time servers issuing time-specific trapdoors at the beginning of each time period. These methods alone only allow the universal disclosure of encrypted documents.

Encrypting a message for a designated receiver and a specific future moment is possible by combining IBE-based unlock methods of [16,27,11] with a traditional public key encryption scheme. Such a composition is especially attractive with the time-based decryption procedure suggested by Boneh, Boyen and Goh ([11]) which consists in using the tree-like structure of Canetti et al. [15] backwards: indeed, it allows recovering past time-specific trapdoors from a current trapdoor. Nevertheless, the root of the tree-like structure of Canetti et al. ([15]) has to correspond to the last time period which implies an upper bound on the lifetime of the system. Unless special precautions are taken, such a composition would also leak information on the release-time of ciphertexts as the hierarchical IBE system of [11] does not have the receiver-anonymity property in the sense of Bellare et al. ([5]).

In this paper, we do not only focus on improving the efficiency of generic constructions. We also aim at providing TRE systems with the property of *release-time confidentiality* according to which ciphertexts do not reveal information to anyone but the intended receiver about their release-time. We also stress that the problem that we address is different from the 'token-controlled public key encryption problem' ([4]) where a sender encrypts a message using a specific token before handing it to a semi-trusted agent (that must communicate with senders who cannot remain anonymous) who stores it until it can be made available to the receiver for completing the decryption.

In our context, a scalable timed-release encryption (TRE) scheme wherein the time server never has to interact with the sender nor the receiver was recently

suggested by Blake and Chan ([8]) who were followed by [28,24] and for different applications by ([19]). In these settings, the sole responsibility of the server is to periodically issue time-specific trapdoors enabling the decryption of ciphertexts encrypted "for the future" ([8,28]) or the hatching of signatures ([19]).

It is to be noted that the scalable TRE solution given by Blake and Chan was not directly supported by a formal security model and only informal security arguments were given in [8] for a scheme that can also be thought of as a particular case of a solution proposed in [1] to tackle with access control problems using pairing based cryptography. We believe that security models considered in [1,24] should be strengthened a little and one of our contributions is to consider a more stringent formal security model for the specific application of non-interactive timed-release encryption. We have to mention that, independently of our work, [28] also considers a security model for authenticated timed-release encryption schemes. We focus here on the mere public key encryption case and we believe our model to be stronger than the one in [28] as well.

In this paper, we also propose a more efficient non-interactive TRE scheme than [8] and [1]. In anonymity enhancing purposes, we then explain how to avoid having to transmit release-times of ciphertexts in clear through insecure channels by hiding them from anyone but their intended recipient and we show how to add this security property to the scheme of [8] for which we also give a security proof in our model.

Both solutions may find other applications than the timed-decryption of digital documents. Similarly to the non-interactive solution of Blake and Chan, ours can be turned into an event-release encryption (ERE) scheme solving the problem of a sender who wishes to send a message that the recipient can only decrypt if a specific event occurs. As an example, we think of the context of a war-correspondent sealing an envelope containing sensitive information with the instruction "to open only if something happens to me". In such a situation, the time server can be turned into a notary that has to verify the occurrence of the prescribed event before issuing a certificate testifying of the event's happening.

Before describing our solutions, we formally define the concept of non-interactive timed-release encryption and we introduce a strong adversarial model which is inspired from the one of certificateless encryption schemes (CLE) ([2,3]). Section 3 then explains why a secure TRE scheme cannot be generically obtained from a secure CLE scheme contrary to what appears at first glance. The new TRE system is presented in section 4 while section 5 explains how to provide our system and the one of Blake and Chan with the newly defined release-time confidentiality property.

## 2 Formal Definition and Adversarial Models

Our model of timed-release encryption schemes assumes that ciphertexts always contain information about their release-time. More precisely, for some $t \in \mathbb{N}$, their last $t$ bits are a label indicating the moment at which their receiver will be allowed to decrypt them.

**Definition 1.** *A TRE scheme is a 5-uple of algorithms:*

**TRE.Setup:** *is a probabilistic algorithm run by a time server to generate system-wide parameters* $\mathtt{params}$ *that include a public key* $\mathtt{TS}_{pub}$ *for which the corresponding private key* $\mathtt{ts}_{priv}$ *is stored in order to be used in all time-specific trapdoor generations.*

**TRE.User-Keygen:** *is a probabilistic algorithm taking as input public parameters* $\mathtt{params}$ *that is run by each end-user to generate a key pair* $(\mathtt{upk}, \mathtt{usk})$*. The public keys are required to have a special form and their validity should be verifiable in polynomial time.*

**TRE.TS-Release:** *is an algorithm run by the server that, given* $\mathtt{ts}_{priv}$ *and a time information* $T \in \{0,1\}^t$*, generates and discloses a specific trapdoor* $s_T$*. The latter's validity should be verifiable in polynomial time given* $T \in \{0,1\}^t$ *and* $\mathtt{TS}_{pub}$*.*

**TRE.Encrypt:** *is a probabilistic algorithm taking as inputs public parameters* $\mathtt{params}$*, a recipient's public key* $\mathtt{upk}$*, a message* $m \in \mathcal{M}$ *and a time information* $T \in \{0,1\}^t$ *to produce a ciphertext* $(C, T) = \mathtt{TRE.Encrypt}(m, \mathtt{upk}, \mathtt{params}, T)$ *that the recipient must be unable to decrypt before knowing* $s_T = \mathtt{TRE.TS\text{-}release}(\mathtt{ts}_{priv}, T)$*.*

**TRE.Decrypt:** *is a deterministic algorithm taking as inputs a ciphertext* $(C, T)$*, parameters* $\mathtt{params}$*, a private key* $\mathtt{usk}$ *and a time-specific trapdoor* $s_T$ *to return a plaintext* $m$ *or a distinguished symbol* $\perp$ *if the ciphertext is not properly formed.*

*For consistency, we impose that* $\mathtt{TRE.Decrypt}(\mathtt{usk}, s_T, \mathtt{params}, (C, T)) = m$ *whenever* $(C, T) = \mathtt{TRE.Encrypt}(m, \mathtt{upk}, \mathtt{params}, T) = m$ *for all messages* $m \in \mathcal{M}$ *if* $s_T = \mathtt{TRE.TS\text{-}release}(\mathtt{ts}_{priv}, T)$*.*

We distinguish two kinds of adversaries. We first consider malicious receivers attempting to gain information on the plaintext before its release-time. Such adversaries do not know the server's private key but can freely choose the public key on which they are challenged in a find-then-guess game. In both stages, they have access to a release-time oracle returning trapdoors for any arbitrary time periods but the (adversarially-chosen) one for which the challenge ciphertext is computed. In a chosen-ciphertext scenario, they are also given access to an oracle decrypting other ciphertexts than the challenge. These adversaries are called chosen-time period and ciphertext attackers (CTCA) in contrast to weaker chosen-time period and plaintext attackers (CTPA).

**Definition 2.** *A TRE scheme is secure against chosen-time period and ciphertext attacks (IND-CTCA) if no probabilistic polynomial time (PPT) attacker has a non-negligible advantage in the following game:*

1. *Given a security parameter* $\mathbf{1}^k$*, the challenger runs* $\mathtt{TRE.Setup}(\mathbf{1}^k)$ *and gives the resulting parameters* $\mathtt{params}$ *(that include the server's public key* $\mathtt{TS}_{pub}$*) to* $\mathcal{A}$ *while* $\mathtt{ts}_{priv}$ *is kept secret.*
2. $\mathcal{A}$ *queries a release-time oracle* $\mathtt{TRE.TS\text{-}release}(.)$ *returning trapdoors* $s_T$ *for arbitrary time periods* $T$ *as well as a decryption oracle* $\mathtt{TRE.Decrypt}(.)$

*which, given a ciphertext* $(C, T)$ *and a receiver's public key* $\mathtt{upk}$ *provided by* $\mathcal{A}$*, generates the decryption of* $C$ *using the trapdoor* $s_T$ *even without knowing the private key* $\mathtt{usk}$ *corresponding to* $\mathtt{upk}$*. At some moment,* $\mathcal{A}$ *outputs messages* $m_0, m_1$*, an arbitrary public key* $\mathtt{upk}^*$ *and a time-period* $T^*$ *that was not submitted to the* $\mathtt{TRE.TS\text{-}release}(.)$ *oracle. She gets the challenge* $(C^*, T^*) = \mathtt{TRE.Encrypt}(m_b, \mathtt{upk}^*, \mathtt{params}, T^*)$*, for a hidden bit* $b \xleftarrow{R} \{0, 1\}$*.*

3. *$\mathcal{A}$ issues new release-time queries for any arbitrary time-period but $T^*$ and decryption queries for any ciphertext but the challenge $(C^*, T^*)$ for the public key $\mathtt{upk}^*$. She eventually outputs a bit $b'$ and wins if $b' = b$. As usual, her advantage is $Adv^{ind\text{-}ctca}_{TR-PKE}(\mathcal{A}) := 2 \times Pr[b' = b] - 1$.*

The above model of security against receivers is seemingly stronger than its counterpart in [28] for which target time periods are fixed by the challenger at the beginning of the game instead of being adaptively chosen by adversaries. When compared to the notion of 'recipient security' defined in [1] or its counterpart in [24], definition 2 also looks stronger as the authors of [1,24] explicitly omitted to provide the attacker with a decryption oracle and argued that such an oracle is useless since the receiver's private key is known to the adversary. Actually, she might still gain useful information by asking for the decryption of ciphertexts $(C, \mathtt{T}^*) \neq (C^*, \mathtt{T}^*)$ for the target time period $\mathtt{T}^*$. That is why, although the challenger does not a priori know any private key except the server's one, we provide the attacker with an oracle that is more powerful than an usual decryption oracle: given a time-information string, a receiver's public key $\mathtt{upk}$ and a ciphertext, it either returns a plaintext or a rejection message even if it does not know the matching private key $\mathtt{usk}$ for $\mathtt{upk}$.

The latter requirement might look too strong in practice but it is to be noted that a similar constraint was imposed by Al-Riyami and Paterson in their security model for certificateless encryption schemes (CLE) ([2,3]). As they did in their context, we can argue here that an adversary has more power if she can obtain the decryption of ciphertexts for receivers's public keys that she simply observes without knowing the matching private key. Besides, since the scheme that we propose in section 4.1 perfectly supports this constraint, we do not believe the latter to be too strong.

Finally, in the model of [1], the challenge key pair $(\mathtt{usk}^*, \mathtt{upk}^*)$ is chosen by the challenger at the outset of the game. Our model does not assume $\mathtt{usk}^*$ to be known to the challenger. It is unclear whether this distinction is of any practical relevance but it seems more natural to allow adversaries to be challenged on any receiver's public key of their choosing without directly revealing the associated private key (which is not needed to compute the challenge ciphertext after all). In fact, the knowledge of $\mathtt{usk}^*$ is not needed in the security proof of our scheme.

In a second definition, we consider the threat of curious servers where attackers know the server's private key but are challenged on a random user's public key for which they are equipped with a decryption oracle.

**Definition 3.** *A TRE scheme is said to be secure against chosen-ciphertext attacks (or IND-CCA secure) if no PPT adversary $\mathcal{A}$ has a non-negligible advantage in the following game:*

1. *Given* $\mathbf{1}^k$, *the challenger* $\mathcal{CH}$ *runs the algorithms* $\texttt{TRE.Setup}(\mathbf{1}^k)$ *and* $\texttt{TRE.User-Keygen}$ *to obtain a list of public parameters* $\texttt{params}$ *and a pair* $(\texttt{upk}, \texttt{usk})$. $\mathcal{CH}$ *gives* $\texttt{params}$, *the server's private key* $\texttt{ts}_{priv}$ *and the public key* $\texttt{upk}$ *to* $\mathcal{A}$ *while the private key* $\texttt{usk}$ *is kept secret.*
2. $\mathcal{A}$ *is given access to a decryption oracle* $\texttt{TRE.Decrypt}(.)$ *which, given a ciphertext* $(C, \texttt{T})$ *and the time-specific trapdoor* $\texttt{s}_T$ *(which is always computable for the adversary who knows* $\texttt{ts}_{priv}$*), returns the decryption of* $C$ *using the private key* $\texttt{usk}$. *At some point, she outputs equal-length messages* $m_0, m_1$ *and a challenge time-period* $\texttt{T}^*$. *She gets a ciphertext* $(C^*, \texttt{T}^*) = \texttt{TRE.Encrypt}(m_b, \texttt{upk}, \texttt{params}, \texttt{T}^*)$, *for* $b \stackrel{R}{\leftarrow} \{0, 1\}$, *computed under the public key* $\texttt{upk}$.
3. $\mathcal{A}$ *issues a new sequence of queries but is prohibited from asking for the decryption of the challenge for the time period* $\texttt{T}^*$. *She eventually outputs a bit* $b'$ *and wins if* $b' = b$. *Her advantage is still defined as* $Adv^{\texttt{ind-cca}}_{TR-PKE}(\mathcal{A}) := 2 \times Pr[b' = b] - 1$.

In the full version of this paper, we establish the security of the Blake-Chan ([8]) scheme in our enhanced security model. The IND-CTCA security is proved under a stronger assumption than its counterpart in a weaker sense in [1].

## 3 Why CLE Does Not Imply TRE

The model of security formalized in definition 2 is reminiscent of the definition of security against Type I adversaries against certificateless encryption scheme (CLE) in that the challenger might have to answer decryption queries on ciphertexts presumably created using a public key for which it does not even know the private key. Besides, the scheme that we describe in section 5.2 bears similarities with a CLE scheme recently proposed in [3] in the same way as the Blake-Chan scheme ([8]) has salient similarities with the CLE scheme described in [2].

Actually, it turns out that some constructions may provide instantiations of both primitives but it is very unlikely that a generic transformation can turn a secure CLE into a secure TRE because of differences between formal models: in CLE schemes, some principal's public key is associated to any identity even though no explicit certificate is used. In contrast, time information strings are never bound to any public key.

It is very tempting to believe that a TRE scheme can be generically obtained from a CLE system by turning the Key Generation Center (KGC) into a time server and transforming the partial key private extraction algorithm (see [2] or [3] for details on certificateless primitives) into a release-time algorithm.

The problems arise when attempting to establish the security of the obtained scheme in the sense of definition 3 assuming that the underlying CLE is secure against malicious KGCs (called Type II adversaries in [2]). In the model of security against a Type II adversary ([2]), the latter is disallowed to replace public keys. Now, in the game of definition 3, consider what happens when the attacker issues a decryption query $(C, \texttt{T})$ for a completely arbitrary time period $\texttt{T}$. In the game that it plays against its own challenger, the challenger of definition 3 is

stuck as it may not replace the public key assigned to the entity of identity $\mathtt{T}$ with the challenge public key $\mathtt{upk}$ since replacement queries are forbidden.

Even worse: when the adversary of definition 3 produces her challenge request $(m_0, m_1, \mathtt{T}^*)$, it is very likely that the challenge public key $\mathtt{upk}$ is not associated to $\mathtt{T}^*$ in the game played by the challenger against its own "certificateless challenger". It comes that, even in the chosen-plaintext scenario, the security of the underlying CLE scheme does not imply the security of the obtained TRE.

On the other hand, if the adversary was challenged on a fixed random user's key pair $(\mathtt{usk}^*, \mathtt{upk}^*)$ provided by the challenger in the game of definition 2 as in the definition of 'receiver security' given in [1], the techniques of Dodis and Katz ([18]) would certainly yield a secure TRE by suitably combining an identity based encryption scheme (IBE) with a traditional public key encryption scheme. Nevertheless, because of the special decryption oracles used in definition 2 where the challenger does not even know adversarially controlled private keys, it is unclear whether the same techniques also apply here.

## 4 An Efficient TRE Construction Using Bilinear Maps

This section presents a new efficient timed-release encryption scheme. It makes use of *bilinear map groups* which are groups $(\mathbb{G}_1, \mathbb{G}_2)$ of prime order $p$ for which there exists a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ satisfying the following properties:

1. Bilinearity: $\forall\ u, v \in \mathbb{G}_1,\ \forall\ a, b \in \mathbb{Z}_p^*$, we have $e(u^a, v^b) = e(u, v)^{ab}$
2. Non-degeneracy: if $g$ generates $\mathbb{G}_1$, then $e(g, g)$ generates $\mathbb{G}_2$
3. Computability: $\forall\ u, v \in \mathbb{G}_1,\ e(u, v)$ can be efficiently computed

The security of our construction is proved to rely on the intractability of the following problem that was introduced in [10].

The *q-Bilinear Diffie-Hellman Inversion Problem* ($q$-BDHIP) consists in, given $(g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}) \in \mathbb{G}_1^{q+1}$, computing $e(g, g)^{1/\alpha} \in \mathbb{G}_2$.

### 4.1 The Scheme

In the new scheme, called $\mathtt{TRE1}$, time-specific trapdoors are signatures computed using a signature scheme independently considered in [9] and [30] unlike the scheme of [8] that uses trapdoors computed according to Boneh et al.'s short signature algorithm ([13]). The $\mathtt{TRE1}$ scheme has similarities with a selective-ID secure IBE that was proven secure without random oracles in ([10]) but its security proofs hold in the random oracle model ([7]). The consistency of the scheme is easy to check as

$$e(X^{rh_1(\mathtt{T})}Y^r, \mathtt{s}_T)^{1/a} = e(g^{a(s+h_1(\mathtt{T}))}, g^{\frac{1}{a(s+h_1(\mathtt{T}))}})^r = e(g, g)^r.$$

**TRE.Setup:** given security parameters $k$ and $k_0$, where $k_0$ is polynomial in $k$, this algorithm outputs a $k$-bit prime number $p$, the description of symmetric bilinear map groups $(\mathbb{G}_1, \mathbb{G}_2)$ of order $p$, a generator $g \in \mathbb{G}_1$ and hash functions $h_1 : \{0,1\}^t \to \mathbb{Z}_p^*$, $h_2 : \mathbb{G}_2 \to \{0,1\}^{n+k_0}$ and $h_3 : \{0,1\}^{n+k_0+t} \times {\mathbb{G}_1}^2 \to \mathbb{Z}_p^*$ for some $t \in \mathbb{N}$. A private key $\mathtt{ts}_{priv} := s \xleftarrow{R} \mathbb{Z}_p^*$ and the corresponding public key $\mathtt{TS}_{pub} := g^s \in \mathbb{G}_1^*$ are also chosen. The public parameters are

$$\mathtt{params} := \{p, k, k_0, \mathbb{G}_1, \mathbb{G}_2, g, \mathtt{TS}_{pub}, e, h_1, h_2, h_3, n, \mathcal{M}, \mathcal{C}\}$$

for plaintext and ciphertext spaces $\mathcal{M} := \{0,1\}^n$ and $\mathcal{C} := \mathbb{G}_1^* \times \{0,1\}^{n+k_0+t}$.

**TRE.TS-release:** uses a time information $T \in \{0,1\}^t$ and the server's private key to return a time-specific trapdoor $\mathtt{s}_T = g^{1/(s+h_1(\mathtt{T}))} \in \mathbb{G}_1^*$.

**TRE.User-Keygen:** this algorithm takes as input $\mathtt{params}$, chooses a private key $\mathtt{usk} := a \in \mathbb{Z}_p^*$ and produces A's public key

$$\mathtt{upk} := \langle X = g^a, Y = \mathtt{TS}_{pub}^a \rangle \in \mathbb{G}_1^* \times \mathbb{G}_1^*$$

**TRE.Encrypt:** to encrypt $m \in \{0,1\}^n$ using the time information $\mathtt{T} \in \{0,1\}^t$ and the public key $\mathtt{upk} = \langle X = g^a, Y = \mathtt{TS}_{pub}^a \rangle$, the sender does the following:

1. Check that $\mathtt{upk}$ is correctly formed by verifying that $e(X, \mathtt{TS}_{pub}) = e(g, Y)$.
2. Choose $x \xleftarrow{R} \{0,1\}^{k_0}$, compute $r = h_3(m||x||\mathtt{T}||\mathtt{upk}) \in \mathbb{Z}_p^*$ and the ciphertext

$$C = \langle c_1, c_2, \mathtt{T} \rangle = \langle X^{r h_1(\mathtt{T})} Y^r, (m||x) \oplus h_2(e(g,g)^r), \mathtt{T} \rangle$$

**TRE.Decrypt:** given $C = \langle c_1, c_2, \mathtt{T} \rangle \in \mathcal{C}$, his private key $a$ and the trapdoor $\mathtt{s}_T$, the receiver computes $(m||x) = c_2 \oplus h_2(e(c_1, \mathtt{s}_T)^{1/a}) \in \{0,1\}^{n+k_0}$ and then $r = h_3(m||x||\mathtt{T}||\mathtt{upk}) \in \mathbb{Z}_p^*$. The message is accepted if $c_1 = X^{r h_1(\mathtt{T})} Y^r$ .

**Fig. 1.** The TRE1 scheme

### 4.2 Efficiency Discussions

If $e(g,g)$ is included among the public parameters, an exponentiation in $\mathbb{G}_2$ and a multi-exponentiation in $\mathbb{G}_1$ are needed for both the sender and the receiver while the latter must also compute a pairing.

TRE1 is thus significantly more efficient than the scheme recently proposed by Blake and Chan ([8]) as no pairing must be computed at encryption. It actually happens to be more practical to encrypt messages with distinct release-times in succession for the same recipient. Indeed, the TRE scheme of [8] requires the sender to compute a pairing that depends on the release-time and, if Alice has to send several ciphertexts with distinct release-times to Bob, she has to compute a new pairing for each encryption. Moreover, TRE1 does not need a special (and much less efficient) hash function mapping strings onto a cyclic group (and it thus benefits from a faster release-time algorithm) while both schemes have similar complexities at decryption.

As for the scheme proposed in [8], the sender has to verify the validity of the public key (in step 1 of the encryption algorithm) to ensure that the recipient will be enabled to decrypt the message. Such a checking is fortunately needed only once at the first use of the key.

### 4.3 Security

We stress on the importance of including the public key upk among the inputs of the hash function $h_3$ because the scheme would be insecure in the game of definition 2 otherwise (as the adversary could turn the challenge into another encryption of the same plaintext for a different public key). In a security analysis, theorems 1 and 2 show that TRE1 is secure in the sense of definitions 2 and 3. The proofs are detailed in the full version of the paper.

**Theorem 1.** *Assume that an IND-CTCA attacker $\mathcal{A}$ has an advantage $\epsilon$ against TRE1 when running a time $\tau$, making $q_{h_i}$ queries to random oracles $h_i$ ($i = 1, 2, 3$) and $q_D$ decryption queries. Then the q-BDHIP can be solved for $q = q_{h_1}$ with a probability*

$$\epsilon' > \frac{1}{(q_{h_1} + q_{h_3})(q_{h_2} + q_{h_3})}(\epsilon - q_{h_3}/2^{k_0-1})(1 - 2^{-k})^{q_D}$$

*within a time $\tau' < \tau + O((q_{h_1}^2 + q_{h_3})\tau_{exp})$ where $\tau_{exp}$ is the maximum of the costs of an exponentiation in $\mathbb{G}_1$ and in $\mathbb{G}_2$.*

**Theorem 2.** *Assume that an IND-CCA attacker $\mathcal{A}$ has an advantage $\epsilon$ against TRE1 when running a time $\tau$, making $q_{h_i}$ queries to random oracles $h_i$ ($i = 1, 2, 3$) and $q_D$ decryption queries. Then the 1-BDHIP can be solved with a probability $\epsilon' > (q_{h_2} + q_{h_3})^{-1}(\epsilon - q_{h_3}/2^{k_0-1})(1 - 2^{-k})^{q_D}$ within a time $\tau' < \tau + O(q_{h_3}\tau_{exp})$ where $\tau_{exp}$ is the maximum time to perform an exponentiation in $\mathbb{G}_1$ and in $\mathbb{G}_2$.*

As TRE1 results from a variant of the first Fujisaki-Okamoto ([20]) transform applied to a simpler version of the scheme (details are given in the full paper), the proofs apply a variant of theorem 3 in ([20]).

### 4.4 Encrypting for Multiple Receivers

Interestingly, the scheme is practical to encrypt messages intended to several recipients *with the same release-time* (encrypting with distinct release-times is forbidden as colluding receivers could decrypt the message without having the appropriate trapdoor): given a plaintext $m$ and public keys $\mathtt{upk}_1 = (X_1, Y_1), \ldots, \mathtt{upk}_N = (X_N, Y_N)$, ciphertexts have the form

$$\langle X_1^{rh_1(\mathtt{T})} Y_1^r, \ldots, X_N^{rh_1(\mathtt{T})} Y_N^r, (m||x) \oplus h_2(e(g,g)^r), \mathcal{L} \rangle$$

where $r = h_3(m||x||\mathtt{T}||\mathtt{upk}_1|| \ldots ||\mathtt{upk}_N)$ and $\mathcal{L}$ is a label indicating how each part of ciphertext is associated to each receiver.

The sender still has no pairing to compute: only a multi-exponentiation per receiver (in addition to an exponentiation in $\mathbb{G}_2$) is needed. The Blake-Chan scheme and its generalization ([1]) do not enjoy this efficiency as one pairing per receiver must be computed.

The security proofs are straightforward adaptations of the proofs of theorems 1 and 2 in a security model which is a simple extension of the one described in section 3: in the extension of definition 2, the adversary outputs a set of $N$ public keys at the end of the find stage whereas, in the counterpart of definition 3, she is challenged a vector of $N$ public keys.

## 5 Adding Release-Time Confidentiality

In the security model considered by Di Crescenzo et al. ([17]), the time server is required to interact with the receiver so that the latter obtains the message if the current time exceeds the release-time but nothing can be learned about the latter by the server.

However, as release-times appended to ciphertexts are transmitted in clear to receivers in their model as in ours, nothing can prevent a spying server (or anyone else) observing release-times of ciphertexts from attempting to gain information on who their recipient could be upon release of the corresponding trapdoor by watching who enquires about it within a reasonably small set of users. Such a threat would hamper the key privacy property ([5]) that `TRE1` could be shown to satisfy in an adapted security model if release-times were scrambled.

We believe that, in order to minimize the server's knowledge about who is talking to whom and enhance the protocol's anonymity, it may be desirable to even preclude such a scenario and guarantee the confidentiality of release-times against anyone but intended recipients who can first unmask a part of the received ciphertext using their private key and learn the release-time before obtaining the corresponding trapdoor. We thus define a new notion called *release-time confidentiality* that captures the inability for the server to decide under which out of two release-times of its choice a given ciphertext was created.

**Definition 4.** *A TRE scheme is said to provide* ***release-time confidentiality*** *(or IND-RT-CCA security) if no PPT adversary $\mathcal{A}$ has a non-negligible advantage in the game below:*

1. *Given $\mathbf{1}^k$, the challenger $\mathcal{CH}$ runs the algorithms* $\mathtt{TRE.Setup}(\mathbf{1}^k)$ *and* `TRE.User-Keygen` *to obtain a list of public parameters* `params` *and a pair* $(\mathtt{upk}, \mathtt{usk})$. *$\mathcal{CH}$ gives* `params`*, the server's private key* $\mathtt{ts}_{priv}$ *and the public key* `upk` *to $\mathcal{A}$ while the private key* `usk` *is kept secret.*
2. *$\mathcal{A}$ is given access to a decryption oracle* $\mathtt{TRE.Decrypt}(.)$ *which, given a ciphertext $(C, T)$ and the time-specific trapdoor $s_T$ (which is always computable for the adversary who knows $\mathtt{ts}_{priv}$), returns the decryption of $C$ using the private key* `usk`*. At some moment, she outputs a plaintext $m^*$ and two time periods $T_0^*, T_1^*$ before getting a challenge $C^* =$* $\mathtt{TRE.Encrypt}(m^*, \mathtt{upk}, \mathtt{params}, T_b^*)$*, for $b \stackrel{R}{\leftarrow} \{0,1\}$.*
3. *$\mathcal{A}$ issues a new sequence of queries but she is of course prohibited from requesting the decryption of $C^*$ under the time periods $T_b^*$. She eventually outputs a bit $b'$ and wins if $b' = b$. Her advantage is $Adv_{TR-PKE}^{\mathtt{ind\text{-}rt\text{-}cca}}(\mathcal{A}) := 2 \times Pr[b' = b] - 1$.*

### 5.1 The TRE1 Case

The TRE1 construction does not provide the confidentiality of release-times as they must be appended to ciphertexts and thus transmitted in clear. However, for applications that would require it, a very simple modification of TRE1 satisfies the new property at the cost of a slight increase in the workload of the sender who has to compute an additional multi-exponentiation while the complexity of the decryption algorithm remains unchanged. The only change is that, instead of being transmitted in clear within the ciphertext, the release-time T is scrambled using a hash value of $c'_1 = g^{r h_1(\mathtt{T})} \mathtt{TS}^r_{pub}$ (obtained from an additional random oracle $h_4$) which is also $c_1^{1/a}$ so that the receiver can first unmask it before obtaining the trapdoor.

In the random oracle model, the modified scheme, called TRE2, has the release-time confidentiality property under the standard Diffie-Hellman assumption in $\mathbb{G}_1$ (in order for this new security notion to rely on the latter assumption, we need to feed hash function $h_2$ with both $c'_1$ and $e(g,g)^r$ in the encryption algorithm) as claimed by theorem 3.

**Theorem 3.** *Assume that an attacker $\mathcal{A}$ has an advantage $\epsilon$ against the release time confidentiality of TRE2 in the sense of definition 4 when running a time $\tau$, making $q_{h_i}$ queries to random oracles $h_i$ ($i = 1, \ldots, 4$) and $q_D$ decryption queries. Then there is an algorithm $\mathcal{B}$ solving the computational Diffie-Hellman problem with a probability*

$$\epsilon' > (\epsilon - q_{h_3}/2^{k_0-1})(1 - 2^{-k})^{q_D}$$

*within a time $\tau' < \tau + O(q_{h_3}\tau_{exp}) + O((2q_{h_3} + q_{h_2} + q_{h_4})\tau_p)$ where $\tau_{exp}$ is the maximum time to perform an exponentiation in $\mathbb{G}_1$ and in $\mathbb{G}_2$ and $\tau_p$ is the cost of a pairing computation.*

### 5.2 Release-Time Confidentiality in the Blake-Chan TRE

A very simple method allows adding the release time confidentiality property to the scheme proposed in [8] at a minimal cost: a single additional exponentiation in $\mathbb{G}_1$ is required at encryption while the decryption operation has essentially the same cost as in the original scheme.

Interestingly, this modification allows proving the security under a weaker assumption than for the original version (details will be given in the full version of the paper): the IND-CTPA security is showed under the bilinear Diffie-Hellman assumption while the IND-CPA and IND-RT-CPA securities both rely on the hardness of the standard Diffie-Hellman problem. As for TRE1 and TRE2, the chosen-ciphertext security is obtained via similar transformations to [20,21].

## 6 Conclusion

We proposed a new stringent security model for non-interactive timed-release encryption schemes and presented a new efficient construction fitting this model.

**TRE.Setup:** given a security parameters $k$, this algorithm chooses a a $k$-bit prime number $p$, symmetric bilinear map groups $(\mathbb{G}_1, \mathbb{G}_2)$ of order $p$, a generator $g \in \mathbb{G}_1$ and hash functions $h_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$, $h_2 : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \{0,1\}^n$ and $h_3 : \mathbb{G}_1 \rightarrow \{0,1\}^t$. It also selects a private key $\mathtt{ts}_{priv} := s \xleftarrow{R} \mathbb{Z}_p^*$ and sets $\mathtt{TS}_{pub} := g^s \in \mathbb{G}_1^*$ as the corresponding public key. The ciphertext space is $\mathcal{C} := \mathbb{G}_1^* \times \{0,1\}^{n+t}$ while the space of plaintexts is $\mathcal{M} := \{0,1\}^n$. The public parameters are then

$$\mathtt{params} := \{k, p, \mathbb{G}_1, \mathbb{G}_2, g, \mathtt{TS}_{pub}, e, h_1, h_2, n, \mathcal{M}, \mathcal{C}\}.$$

**TRE.TS-release:** given $\mathtt{T} \in \{0,1\}^t$, the server discloses a trapdoor $\mathtt{s}_T = h_1(\mathtt{T})^s$.

**TRE.User-Keygen:** this algorithm takes as input $\mathtt{params}$, chooses a private key $\mathtt{usk} := a \in \mathbb{Z}_p^*$ and produces A's public key $\mathtt{upk} := X = g^a \in \mathbb{G}_1^*$.

**TRE.Encrypt:** to encrypt $m \in \{0,1\}^n$ for the time period $T \in \{0,1\}^t$ and the public key $\mathtt{upk} = X = g^a$, the sender chooses $r \xleftarrow{R} \mathbb{Z}_p^*$ and the ciphertext is

$$C = \langle c_1, c_2, c_3 \rangle = \langle g^r, m \oplus h_2(X^r || e(\mathtt{TS}_{pub}, h_1(\mathtt{T}))^r), \mathtt{T} \oplus h_3(X^r) \rangle$$

**TRE.Decrypt:** given a ciphertext $C = \langle c_1, c_2, c_3 \rangle \in \mathcal{C}$, a private key $a \in \mathbb{Z}_p^*$, the receiver computes $c_1' = c_1^a \in \mathbb{G}_1^*$ to obtain $\mathtt{T} = c_3 \oplus h_3(c_1') \in \{0,1\}^t$ and recover the plaintext $m = c_2 \oplus h_2(c_1' || e(c_1, \mathtt{s}_T)) \in \{0,1\}^n$ upon release of $\mathtt{s}_T$.

**Fig. 2.** The BC-TRE2 scheme

We also explained how to enhance the anonymity of ciphertexts at a minimum cost in our scheme as in Blake and Chan's one in accordance with a new formally defined security property.

## References


1. S.-S. Al-Riyami , J. Malone-Lee, N.P. Smart, *Escrow-Free Encryption Supporting Cryptographic Workflow*, available from http://eprint.iacr.org/2004/258.
2. S.-S. Al-Riyami , K.G. Paterson, *Certificateless Public Key Cryptography*, in Advances in Cryptology - Asiacrypt'03, LNCS 2894, pp. 452–473, Springer, 2003.
3. S.S. Al-Riyami , K.G. Paterson, *CBE from CL-PKE: A Generic Construction and Efficient Schemes* , in proc. of PKC'05, LNCS 3386, pp. 398–415, Springer, 2005.
4. J. Baek, R. Safavi-Naini, W. Susilo, *Token-Controlled Public Key Encryption,* to appear in proc. of ISPEC'05, LNCS series, 2005.
5. M. Bellare, A. Boldyreva, A. Desai, D. Pointcheval, *Key-Privacy in Public-Key Encryption*, in Advances in Cryptology - Asiacrypt'01, LNCS 2248, pp. 566–582. Springer, 2001.
6. M. Bellare, S. Goldwasser, *Encapsulated key-escrow*, $4^{th}$ ACM Conference on Computer and Communications Security, 1997.
7. M. Bellare, P. Rogaway, *Random oracles are practical: A paradigm for designing efficient protocols*, $1^{st}$ ACM Conference on Computer and Communications Security, pp. 62-73, 1993.
8. I. Blake, A.-C.-F. Chan, *Scalable, Server-Passive, User-Anonymous Timed Release Public Key Encryption from Bilinear Pairing*, available from http://eprint.iacr.org/2004/211/, 2004.

9. D. Boneh, X. Boyen, *Short Signatures Without Random Oracles*, in Advances in Cryptology - Eurocrypt'04, LNCS 3027, Springer, pp. 56–73, 2004.
10. D. Boneh, X. Boyen, *Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles*, in Advances in Cryptology - Eurocrypt'04, LNCS 3027, Springer,pp. 223–238, 2004.
11. D. Boneh, X. Boyen, E.-J. Goh, *Hierarchical Identity Based Encryption with Constant Size Ciphertext* , available at http://eprint.iacr.org/2005/015.
12. D. Boneh, M. Franklin, *Identity Based Encryption From the Weil Pairing*, in Advances in Cryptology - Crypto'01, LNCS 2139, pp. 213–229, Springer, 2001.
13. D. Boneh, B. Lynn, H. Shacham, *Short signatures from the Weil pairing*, in Advances in Cryptology - Asiacrypt'01, LNCS 2248, pp. 514–532, Springer, 2001.
14. D. Boneh, M. Naor, *Timed Commitments*, Advances in Cryptology - Crypto'00, LNCS 1880, pp. 236–254, Springer, 2000.
15. R. Canetti, S. Halevi, J. Katz, *A Forward Secure Public Key Encryption Scheme*, Advances in Cryptology - Eurocrypt'03, LNCS 2656, pp. 254–271, Springer, 2003.
16. L. Chen, K. Harrison, N. Smart, D. Soldera, *Applications of Multiple Trust Authorities in Pairing Based Cryptosystems*, in Infrasec'02, LNCS 2437, pp. 260–275, Springer, 2002.
17. G. Di Crescenzo, R. Ostrovsky, S. Rajagopalan, *Conditional Oblivious Transfer and Timed-Release Encryption*, in Advances in Cryptology - Eurocrypt'99, LNCS 1592, pp. 74–89, Springer, 1999.
18. Y. Dodis, J. Katz, *Chosen-Ciphertext Security of Multiple Encryption*, in TCC'05, LNCS 3378, pp. 188–209, Springer, 2005.
19. Y. Dodis, D.-H. Yum, *Time Capsule Signatures*, to appear in proc. of Financial Cryptography 2005, LNCS series, 2005.
20. E. Fujisaki, T. Okamoto, *How to Enhance the Security of Public-Key Encryption at Minimum Cost*, in proc. of PKC'99, LNCS 1560, pp. 53–68. Springer, 1999.
21. E. Fujisaki and T. Okamoto, *Secure integration of asymmetric and symmetric encryption schemes*, in Advances in Cryptology - Crypto'99, LNCS 1666, pp. 537–554. Springer, 1999.
22. J. Garay, M. Jakobsson, *Timed-Release of Standard Digital Signatures*, in Financial Crypto'02, LNCS 2357, pp. 168–182, Springer, 2002.
23. J. Garay, C. Pomerance, *Timed Fair Exchange of Standard Signatures*, in Financial Crypto'03, LNCS 2742, pp. 190–207, Springer, 2003.
24. Y. H. Hwang, D. H. Yum, P. J. Lee *Timed-Release Encryption with Pre-open Capability and its Application to Certified E-mail System*, to appear in ISC'05, LNCS series, 2005.
25. W. Mao, *Timed-Release Cryptography*, in Selected Areas in Cryptography'01, LNCS 2259, pp. 342–357, Springer, 2001.
26. T. May, *Time-release crypto*, manuscript, February 1993.
27. M.C. Mont, K. Harrison. M. Sadler, *The HP time vault service: Innovating the way confidential information is disclosed at the right time*, in 12th International World Wide Web Conference, pp. 160–169, ACM Press, 2003.
28. I. Osipkov, Y. Kim, J.-H. Cheon, *Timed-Release Public Key Based Authenticated Encryption*, available from http://eprint.iacr.org/2004/231.
29. R. Rivest, A. Shamir, D.A. Wagner, *Time-lock puzzles and timed-release crypto*, MIT LCS Tech. Report MIT/LCS/TR-684, 1996.
30. F. Zhang, R. Safavi-Naini, W. Susilo, *An Efficient Signature Scheme from Bilinear Pairings and Its Applications*, in proc. of PKC'04, LNCS 2947, pp. 277–290, 2004.