PAIRING BASED TIMED-RELEASE CRYPTOGRAPHY

K.Chalkias F.Baldimtsi D.Hristu-Varsakelis
G.Stephanides

Computational Systems and Software Engineering Laboratory,
Department of Applied Informatics,
University of Macedonia,
156 Egnatia St.,
Thessaloniki, Greece
{*chalkias, foteini*}*@java.uom.gr* {*dcv, steph*}*@uom.gr*

Identity Based Encryption Workshop, NIST 2008

Outline

Sending Information into the Future

Timed-Release Cryptography (TRC)

"the encryption of confidential data so that the resulting ciphertext cannot be decrypted by anyone, including the designated recipient(s), until a predetermined future time"

Methods for TRE

Methods for TRE

1. The classic method: the easiest way to provide TRE is to encrypt a message and then send the decryption key at the desired time in the future.

Methods for TRE

1. The classic method: the easiest way to provide TRE is to encrypt a message and then send the decryption key at the desired time in the future.
2. Time-Lock Puzzles (TLP): the receiver needs to perform some non-parallelizable computation without stopping in order to recover a message

Methods for TRE

1. The classic method: the easiest way to provide TRE is to encrypt a message and then send the decryption key at the desired time in the future.
2. Time-Lock Puzzles (TLP): the receiver needs to perform some non-parallelizable computation without stopping in order to recover a message
3. Trusted Agents (TA): they are based on a trusted third-party (often referred to as the 'time-server' or TTP) whose function is to provide a common and absolute time reference to users.

For and Againsts of each TRE method

For and Againsts of each TRE method

1. The classic method is impractical, because the receiver MUST be online at the selected time instant (No guaranty).

For and Againsts of each TRE method

1. The classic method is impractical, because the receiver MUST be online at the selected time instant (No guaranty).
2. TLPs
   1. they are third-party independent.
   2. they can only be practical for short period of times (e.g. as a time-delay function)
   3. they cannot guarantee precise timing of information release
   4. they put immense computational overhead on the receiver's CPU
   5. the total time depends on the time at which the decryption process is started

For and Againsts of each TRE method

1. The classic method is impractical, because the receiver MUST be online at the selected time instant (No guaranty).
2. TLPs
   1. they are third-party independent.
   2. they can only be practical for short period of times (e.g. as a time-delay function)
   3. they cannot guarantee precise timing of information release
   4. they put immense computational overhead on the receiver's CPU
   5. the total time depends on the time at which the decryption process is started
3. TAs
   1. they provide absolute release time
   2. there exist practical and efficient constructions
   3. they require a third entity

TRE from Trusted Agents: Categories

TRE from Trusted Agents: Categories

1. Simple Time-Servers
    1. Key Escrow Agents
    2. Key Pair Generators (they publish a keypair for the desired time instants)

TRE from Trusted Agents: Categories

1. Simple Time-Servers
   1. Key Escrow Agents
   2. Key Pair Generators (they publish a keypair for the desired time instants)
2. Passive Time-Servers
   1. TRE based on Quadratic Residues
   2. TRE based on Bilinear Pairings

Why Pairing-Based TRE?

Why Pairing-Based TRE?

1. Simple,Passive Time-Servers: they provide a common time reference by periodically releasing unforgeable, time-embedded information, which will be used to decrypt timed-release ciphertexts

Why Pairing-Based TRE?

1. Simple,Passive Time-Servers: they provide a common time reference by periodically releasing unforgeable, time-embedded information, which will be used to decrypt timed-release ciphertexts
2. No need for additional signatures, trapdoors are selfsigned

Why Pairing-Based TRE?

1. Simple,Passive Time-Servers: they provide a common time reference by periodically releasing unforgeable, time-embedded information, which will be used to decrypt timed-release ciphertexts
2. No need for additional signatures, trapdoors are selfsigned
3. Sender Anonymous - no need to interact with the server

Why Pairing-Based TRE?

1. Simple,Passive Time-Servers: they provide a common time reference by periodically releasing unforgeable, time-embedded information, which will be used to decrypt timed-release ciphertexts
2. No need for additional signatures, trapdoors are selfsigned
3. Sender Anonymous - no need to interact with the server
4. Scalable: it can be extended for use with multiple TAs

Why Pairing-Based TRE?

1. Simple,Passive Time-Servers: they provide a common time reference by periodically releasing unforgeable, time-embedded information, which will be used to decrypt timed-release ciphertexts
2. No need for additional signatures, trapdoors are selfsigned
3. Sender Anonymous - no need to interact with the server
4. Scalable: it can be extended for use with multiple TAs
5. Less communication cost than QR-TRE

Pairing-Based TRE

Pairing-Based TRE

1. Directly from IBE: Encryption is possible even before the receiver gets her private key!

Pairing-Based TRE

1. Directly from IBE: Encryption is possible even before the receiver gets her private key!
2. Secret Sharing: Encrypt the first part of the message with an IBE scheme (for ID = time) and the second part with any encryption scheme targeted to the receiver.

Pairing-Based TRE

1. Directly from IBE: Encryption is possible even before the receiver gets her private key!
2. Secret Sharing: Encrypt the first part of the message with an IBE scheme (for ID = time) and the second part with any encryption scheme targeted to the receiver.
3. Specific TRE schemes:
   1. based on BLS short signatures
   2. based on BB short signatures
   3. based on hierarchical IBE

Applications

Applications

1. E-Voting Systems

Applications

1. E-Voting Systems
2. Sealed Bid e-Auctions

Applications

1. E-Voting Systems
2. Sealed Bid e-Auctions
3. E-contests

Applications

1. E-Voting Systems
2. Sealed Bid e-Auctions
3. E-contests
4. Online Gambling and Games

Applications

1. E-Voting Systems
2. Sealed Bid e-Auctions
3. E-contests
4. Online Gambling and Games
5. Release of Electronic Documents

Applications

1. E-Voting Systems
2. Sealed Bid e-Auctions
3. E-contests
4. Online Gambling and Games
5. Release of Electronic Documents
6. Payments Schedules

Applications

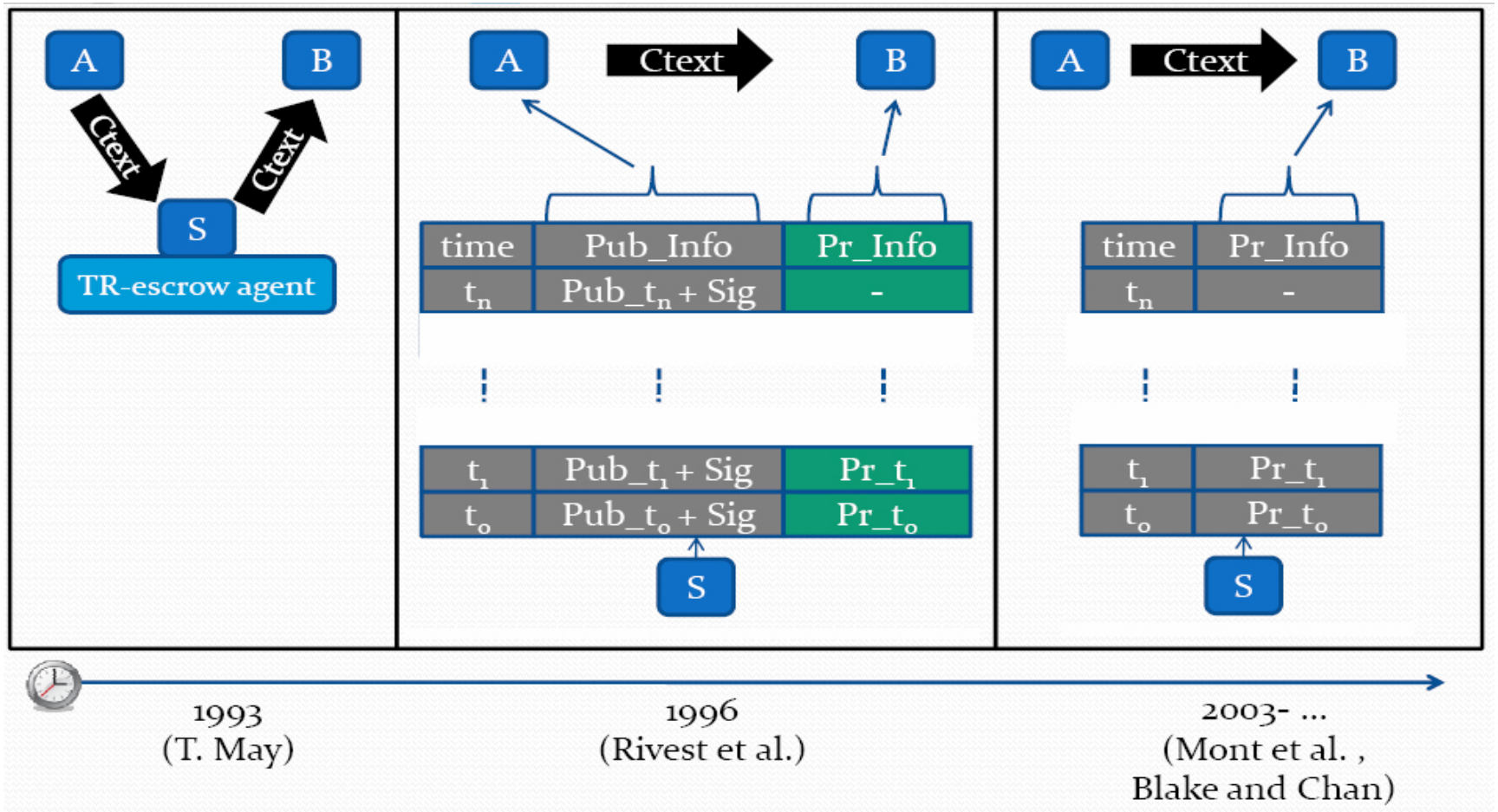
1. E-Voting Systems
2. Sealed Bid e-Auctions
3. E-contests
4. Online Gambling and Games
5. Release of Electronic Documents
6. Payments Schedules
7. Contract Signing

Applications

1. E-Voting Systems
2. Sealed Bid e-Auctions
3. E-contests
4. Online Gambling and Games
5. Release of Electronic Documents
6. Payments Schedules
7. Contract Signing
8. SMS and e-mails

Applications

1. E-Voting Systems
2. Sealed Bid e-Auctions
3. E-contests
4. Online Gambling and Games
5. Release of Electronic Documents
6. Payments Schedules
7. Contract Signing
8. SMS and e-mails
9. and more...


A
B
Ctext
Ctext
S
TR-escrow agent
A
Ctext
B
time
Pub_Info
Pr_Info
t_n
Pub_t_n + Sig
-
t_1
Pub_t_1 + Sig
Pr_t_1
t_o
Pub_t_o + Sig
Pr_t_o
S
A
Ctext
B
time
Pr_Info
t_n
-
t_1
Pr_t_1
t_o
Pr_t_o
S
1993
(T. May)
1996
(Rivest et al.)
2003- ...
(Mont et al. ,
Blake and Chan)

Preliminaries (1)

- $\mathbb{G}_1$: abelian additive finite group of prime order q
- $\mathbb{G}_2$: abelian multiplicative cyclic group of the same order
- P: generator of $\mathbb{G}_1$
- H_n: secure hash function
- $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$: bilinear pairing

Preliminaries (1)

- $\mathbb{G}_1$: abelian additive finite group of prime order q
- $\mathbb{G}_2$: abelian multiplicative cyclic group of the same order
- P: generator of $\mathbb{G}_1$
- H_n: secure hash function
- $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$: bilinear pairing

Bilinear Pairings

- *Bilinear:* $\hat{e}(aP, bQ) = \hat{e}(abP, Q) = \hat{e}(P, abQ) = \hat{e}(P, Q)^{ab}$ *for all* $P, Q \in \mathbb{G}_1$ *and* $a, b \in \mathbb{Z}_q^*$
- *Non-degenerate: there exists* $P, Q \in \mathbb{G}_1$ *such that* $\hat{e}(P, Q) \neq \mathbb{G}_1$
- *Efficient: there exists an efficient algorithm to compute the bilinear map*

Preliminaries (2)

Discrete Logarithm Problem

Given $Q, R \in \mathbb{G}_1$ find an integer $a \in \mathbb{Z}_q^*$ such that $R = aQ$.

Preliminaries (2)

Discrete Logarithm Problem

Given $Q, R \in \mathbb{G}_1$ find an integer $a \in \mathbb{Z}_q^*$ such that $R = aQ$.

Computational Diffie-Hellman Problem

Given $Q \in \mathbb{G}_1$, aQ, bQ for some unknowns $a, b \in Z_q^*$, compute abQ.

Preliminaries (2)

Discrete Logarithm Problem

Given $Q, R \in \mathbb{G}_1$ find an integer $a \in \mathbb{Z}_q^*$ such that $R = aQ$.

Computational Diffie-Hellman Problem

Given $Q \in \mathbb{G}_1$, aQ, bQ for some unknowns $a, b \in Z_q^*$, compute abQ.

Bilinear Diffie-Hellman Problem

Given $Q \in \mathbb{G}_1$, aQ, bQ and cQ for some unknowns $a, b, c \in Z_q^*$, compute $\hat{e}(Q, Q)^{abc}$.

A TRE Example Scheme

HCS protocol

Sender	**Receiver** $(b, B = bP)$	**Server** $(s, S = sP)$
		$T = H_1(t)$ $s_{kt} = sT$
$r \in_R \mathbb{Z}_q^*$ $T = H_1(t)$ $Q = rT$ $K = \hat{e}(S, Q) = \hat{e}(sP, rT) = \hat{e}(P, T)^{rs}$ $c_1 = rB = rbP$ $c_2 = m \oplus H_2(K)$ $\xrightarrow{\langle c_1, c_2, t \rangle}$		
	$R = b^{-1}c_1 = b^{-1}brP = rP$ R can be precomputed $K = \hat{e}(R, s_{kt}) = \hat{e}(rP, sT) = \hat{e}(P, T)^{rs}$ $m = H_2(K) \oplus c_2$	$\xleftarrow{s_{kt}}$[at release time]

BLS-Based TRE

BLS short signature scheme proposed by Boneh, Lynn and Shacham in '01. Security is proven under the random-oracle model. If P is a generator of $\mathbb{G}_1$, $H : \{0,1\}^* \mapsto \mathbb{G}_1$ and $(a, A = aP)$ is Alice's keypair, then the $BLS - Sig$ of Alice in the message m is defined as:

Sign(m): $\sigma = H(m)^a$.

Ver(σ, m): $\hat{e}(P, \sigma) \stackrel{?}{=} \hat{e}(A, H(m))$

Protocols

- Blake and Chan (2004) The first server-passive PB-TRE
- Hwang et al. (2005) + Message PreOpening
- Dent and Tang (2006) Efficient + Message PreOpening
- Hristu et al. (2007) Efficient + Multiple time-servers
- Osipkov et al. (2004) Authenticated TRE
- Cheon et al. (2006) Similar to Osipkov et al.

BB-Based TRE

BB short signature scheme proposed by Boneh and Boyen '04, and Zhang et al '04. This scheme was initially used in the selective-ID secure IBE which was proven to be secure without random oracles. If P is a generator of $\mathbb{G}_1$, $h : \{0,1\}^* \mapsto \mathbb{Z}_q^*$ and $(a, A = aP)$ is Alice's keypair, then the *BBSig* of Alice in the message m is defined as:
Sign(m): $\sigma = (a + h(m))^{-1} P$.
Ver(σ, m): $\hat{e}(A + h(m)P, \sigma) \stackrel{?}{=} \hat{e}(P, P)$
Protocols

- Yoshida et al. (2004/05) Backward Trapdoor Recovery
- Cathalo et al. (2005) Pre-Computations + Confidentiality of Release Time
- Chalkias et al. (2007) Efficient + Simple DH Keys

- Nali et al. (2006): it can be used to efficiently handle large user communities which are hierarchically structured, (e.g. employees of a large corporation)

- Nali et al. (2006): it can be used to efficiently handle large user communities which are hierarchically structured, (e.g. employees of a large corporation)
- Chow et al. (2008): the first TRE in the Standard Model

Requirements:

- If the signer wants, she can make her time capsule signature effective before the pre-defined time t

Requirements:

- If the signer wants, she can make her time capsule signature effective before the pre-defined time t
- The recipient of 'future signature' can verify right away that the signature will become valid no later than at time t

Requirements:

- If the signer wants, she can make her time capsule signature effective before the pre-defined time t
- The recipient of 'future signature' can verify right away that the signature will become valid no later than at time t
- Time-Server need not contact any user at any time, and in fact does not need to know anything about the PKI employed by the users

Requirements:

- If the signer wants, she can make her time capsule signature effective before the pre-defined time t
- The recipient of 'future signature' can verify right away that the signature will become valid no later than at time t
- Time-Server need not contact any user at any time, and in fact does not need to know anything about the PKI employed by the users
- Signatures completed by the signer before time t are indistinguishable from the ones completed using the Time Server at time t

Protocols [DY05][ZCLWQ06][LQ07]

Additional Properties

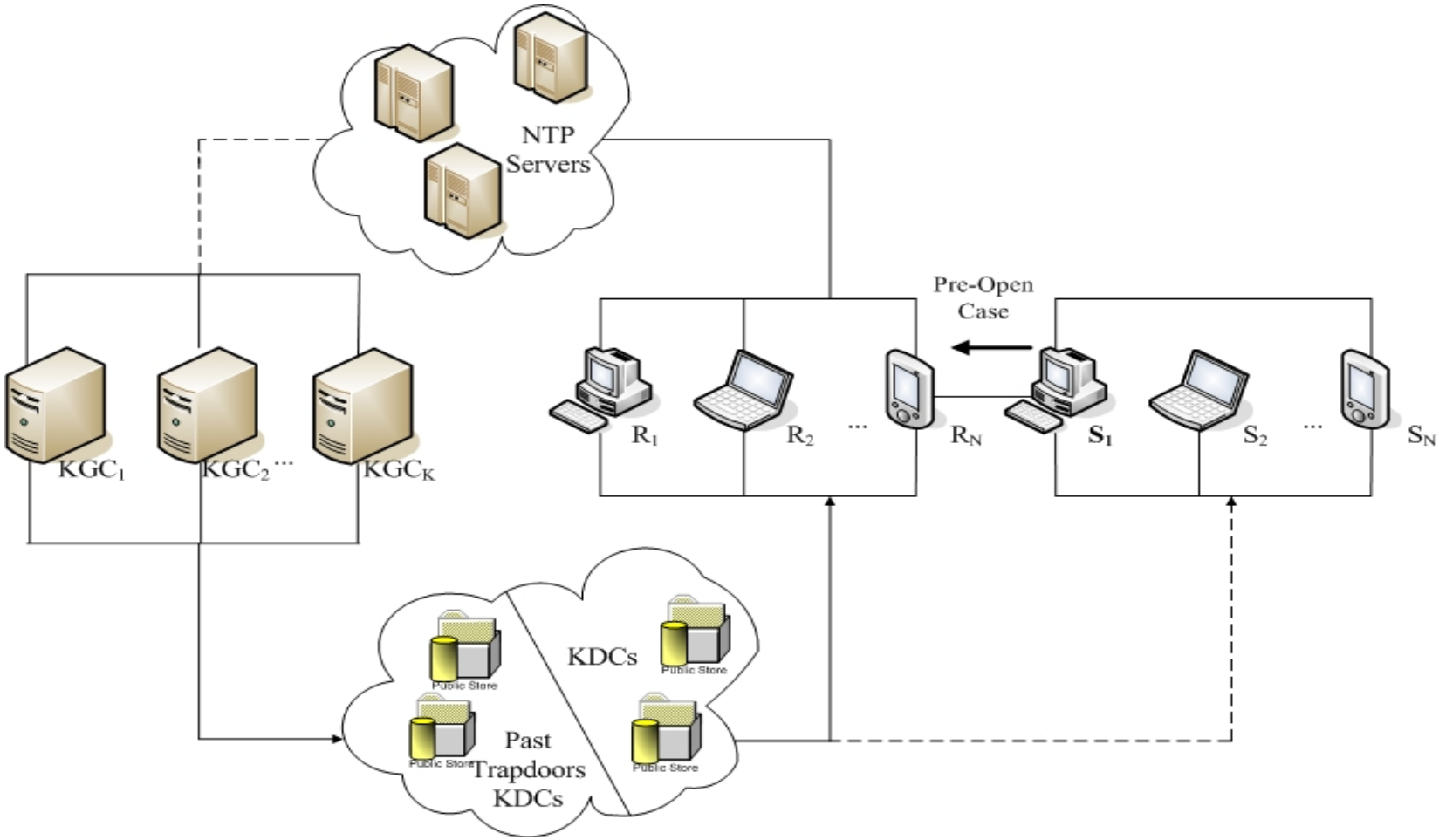
- Multiple time-server support: it should be possible to support the use of multiple time-servers when encrypting/decrypting, in order to eliminate, or at least reduce, the possibility of collusion between the receiver and an unscrupulous time-server.
- Pre-open capability: a sender should have the option to allow early decryption of a message by sending to the receiver a trapdoor key (different from the one to be issued by the time-server) before the designated time.
- Confidentiality of release time: there should be an option to "hide" the disclosure time.
- Public part: an application may require that part of the message be public, i.e., viewable by anyone at any time.

Current TRE Infrastructures

Although by now there exist a significant number of modern TRE approaches, each with its own desirable features with respect to security, anonymity and other properties, there has been little work on the infrastructure(s) which will be required in order to implement the theoretical work.

- [CDBN06]
- [MHS03]
- [CBHS08]

The model


NTP
Servers
KGC1
KGC2
KGCK
Pre-Open
Case
R1
R2
RN
S1
S2
SN
KDCs
Public Store
Past
Trapdoors
KDCs

IB-TRE and CL-TRE

IB-TRE and CL-TRE

1. IB-TRE: It is possible to construct an efficient IB-TRE where the Trusted Authority = Time-Server [BC'04]. However, this system is insecure against malicious time-servers.
   Possible Solutions:
   1. Trusted Authority $\neq$ Time Server
   2. Create a multi-server edition of the protocol
2. CL-TRE:
   1. A CL-TRE scheme can be constructed generically by combining any CLE scheme with any IBE scheme. Using secret-sharing techniques, a sender can split a message and encrypt one part under the receiver's CLE key and the other part under the ID that corresponds to the desired date.
   2. Efficient Concrete Schemes e.g., [CHS'08].

*In a CLE setting one can protect against malicious KGCs without introducing additional time-servers.

E-voting

The process of holding an election electronically, with ballots cast securely and secretly.

- lower error rates in vote counting
- no need for physical voter presence
- lower cost

Crucial security requirements, such as vote accuracy, democracy, verifiability, voter privacy and double-voting detection.

TRC in E-voting

- prevent the early opening of electronically-case votes
- avoid election fraud - all parties involved do not have access to the results until a specific, predefined time in the future
- prevent communications bottlenecks that would occur if all votes had to be cast "just in time"
- secrecy
- voter anonymity
- higher security - multiple time servers

Sealed-bid E-auctions

A negotiation mechanism where sellers and buyers intend to come to an agreement on the transaction of a commodity. Each bidder submits a sealed bid stating how much he is willing to pay and the highest (or the second, or third highest - depending on the method used) bid wins the auction.

- secrecy - one should be able to view the bids before the bidding period has ended

Timed release encryption would prevent these problems from arising by making it difficult for anyone to view the bids before the end of the auction, thus enforcing honesty among participants.

Others (1)

- **E-contests** simultaneous access to the challenge problem, despite possible network congestion or delivery delays - with TRE every participant receives the challenge well before the contest starts

Others (1)

- **E-contests** simultaneous access to the challenge problem, despite possible network congestion or delivery delays - with TRE every participant receives the challenge well before the contest starts
- **Online gambling and Games** fair game, results on random bases unable to be influenced or manipulated by the entity or other players

Others (1)

- **E-contests** simultaneous access to the challenge problem, despite possible network congestion or delivery delays - with TRE every participant receives the challenge well before the contest starts
- **Online gambling and Games** fair game, results on random bases unable to be influenced or manipulated by the entity or other players
- **Release of electronic documents** document not revealed until the appointed time (e.g., memoirs, wills, business plans, strategic decisions)

Others (1)

- **E-contests** simultaneous access to the challenge problem, despite possible network congestion or delivery delays - with TRE every participant receives the challenge well before the contest starts
- **Online gambling and Games** fair game, results on random bases unable to be influenced or manipulated by the entity or other players
- **Release of electronic documents** document not revealed until the appointed time (e.g., memoirs, wills, business plans, strategic decisions)
- **Payment schedules** specific dates of payements

Others (2)

- **Contract Signing** two or more remote and mutually suspicious parties wishing to exchange signatures on a contact

Others (2)

- **Contract Signing** two or more remote and mutually suspicious parties wishing to exchange signatures on a contact
- **Time-stamping** a sequence of characters, denoting the date and/or time at which a certain event occurred. Trapdoors embodied to any application as an unimpeachable time reference.

Others (2)

- **Contract Signing** two or more remote and mutually suspicious parties wishing to exchange signatures on a contact
- **Time-stamping** a sequence of characters, denoting the date and/or time at which a certain event occurred. Trapdoors embodied to any application as an unimpeachable time reference.
- **SMS and e-mail**

Question

Thank you for your time...
Q & A