Efficient Unidirectional Proxy Re-Encryption

New York University

Sherman Chow

Jian Weng

Yanjiang Yang

Robert Deng

Jinan University

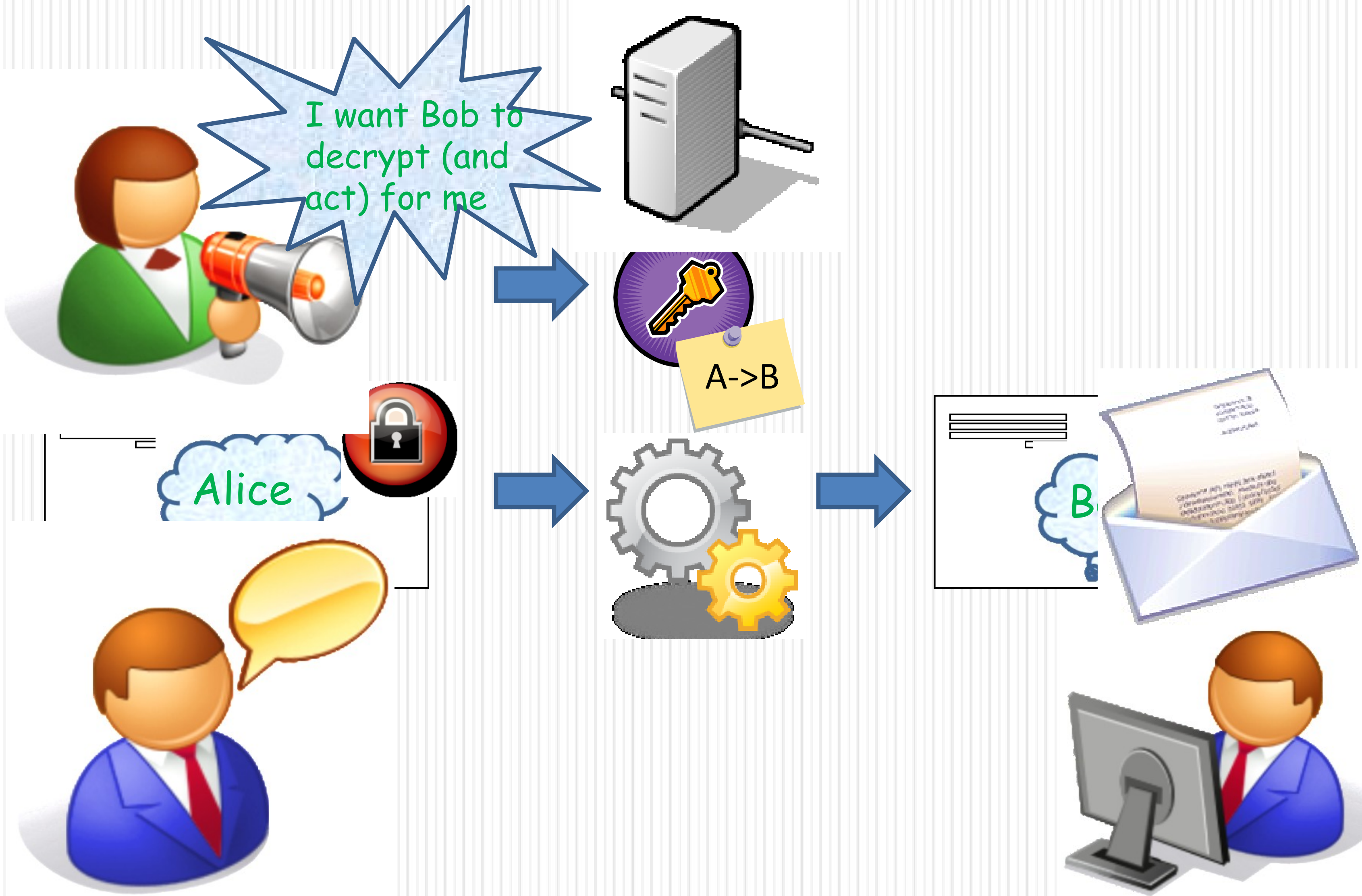
AFRICACRYPT 2010

Presented by Alfredo Rial, 05/05/2010

Encrypted Email Forwarding

- You are now away for Africacrypt.
- You want to forward your incoming emails to your secretary.
- You give your private key to your secretary?
- You deploy your private key on your machine?

Proxy Re-Encryption (PRE)



Applications

- Encrypted email forwarding
  - Blaze, Bleumer, Strauss 98
- Law enforcement
  - Ivan, Dodis 03
- Digital rights management
  - Apple iTunes
- Distributed file storage systems
- Outsourced filtering of encrypted spam
  - Ateniese, Fu, Green, Hohenberger 06

Properties

- “Single-hop”
- Unidirectional
  - A → B does not mean B → A
- Collusion-resistance
  - Basic: proxy and delegatee can’t recover the private key of delegator “in full”
  - This talk: can’t compromise the security of delegator in “any meaningful way”

Summary of PRE Schemes

Schemes	Uni/Bi dir.	Security	RO-free	Pairing -free	Collusion resistant
[AFGH06]	->	CPA	👎	👎	☺
[HRSV07]	->	CCA	☺	👎	☺
[CH07]	<->	CCA	☺	👎	👎
[LV08]	->	RCCA	☺	👎	☺
[LV08-T]	->	CPA	☺	👎	☺
[DWLC08]	<->	CCA	👎	☺	👎
[SC09]	->	CCA?	👎	☺	👎
[ABH09]	->	CPA	☺	👎	☺
Ours	->	CCA	👎	☺	☺

Why pairing?

- Unidirectional $rk_{i\ ->j} = g\text{^}(sk_j\ /\ sk_i)$
- Libert-Vergnaud 08: $e(rk_{i\ ->j}, (pk_i)^r) = e(g, pk_j)^r$
  - Use $(1\ /\ sk_j)$ to get the padding $e(g, g)^r$
- Use pairing $e()$ for ciphertext validity verification
  - only transforms valid ciphertext for CCA concern

Our Contributions

- Definition:
  - A new security model for PRE built from the "token-controlled encryption" approach
- Attack:
  - CCA of a PRE scheme by Shao-Cao in PKC '09
  - Can fix it, but still relatively inefficient
  - Decisional Diffie-Hellman over $\mathbf{Z}^*_{N^2}$
- Construction:
  - PRE realized without pairing
  - Efficient PRE with simple design

Framework

- KeyGen(), Enc(pk, m), Dec(sk, C)

- $rk_{i \rightarrow j}$ ← ReKeyGen(sk_i, pk_j)
- C_j ← ReEnc($rk_{i \rightarrow j}$, C_i)

Our Model

- Knowledge of Secret Key assumption
  - As in [CH07, LV08]
- Random oracle
- CCA instead of RCCA
  - E.g., [LV08] tolerates a “harmless mauling” of the challenge ciphertext
  - At the expense of additional constraint on the re-encryption key that can be compromised
- Collusion: returns a combination of the delegator, delegatee and proxy’s secrets

Game Template

- Setup generates lists PK_{good} (honest user's keys) and Pk_{corr} (corrupted)
  - Gives all *PK*s and SK_{corr} to adversary *Adv*
- Decryption oracle: ODec
- Transformation Key oracle: OReK
- Re-Encryption oracle: OReE
- *Adv* chooses m_0, m_1, pk_{i*} in PK_{good}

Original Ciphertext Security

- Challenge $C^* = \text{Enc}(pk_{i^*}, m_b)$
- *Adv* can't re-encrypt the challenge to a compromised user pk_j in Pk_{corr}
- No $\text{OReK}(pk_{i^*}, pk_j)$
- If *Adv* issued $\text{OReE}(pk_i, C_i, pk_j)$
- Or if *Adv* issued $\text{ODec}(pk_i, C_i)$
- (pk_i, C_i) can't be derived from (pk_{i^*}, C^*)

Derivative for CCA Security

- If *Adv* has issued OReE(*pk*, *pk'*, *C*) and obtained *C'*, then (*pk'*, *C'*) is a derivative of (*pk*, *C*)
- If *Adv* has issued OReK(*pk*, *pk'*) and obtained *rk*, then (*pk'*, ReEnc(*rk*, *C*)) is a derivative of (*pk*, *C*)
- Adopted from RCCA-based definition

Transformed Ciphertext

- $C^* = \mathrm{ReEnc}(rk_{i' \to i^*}, \mathrm{Enc}(pk_{i'}, m_b))$
  - Adv can also specify the delegator $pk_{i'}$
- $\mathrm{ODec}(pk_{i^*}, C^*)$ is not allowed
- If $pk_{i'}$ in Pk_{corr}, would not return $rk_{i' \to i^*}$
- On the other hand, if *Adv* got $rk_{i' \to i^*}$, Adv cannot choose $pk_{i'}$ as the delegator
- This is weaker than [LV08], but …

Constraints in Our Definition

- $C^* = \mathrm{ReEnc}(rk_{i' \to i^*}, \mathrm{Enc}(pk_{i'}, m_b))$
- Both $sk_{i'}$ (delegator) and $rk_{i' \to i^*}$ (proxy) are compromised.
- *Adv* may have obtained the original ciphertext $\mathrm{Enc}(pk_{i'}, m_b)$ and use $sk_{i'}$ to decrypt trivially
- What if they were initially honest and *erased* the original ciphertext?
- *Adv* may capture the ciphertext by itself

Nontransformable Ciphertext

- We only talked about transformed ciphertext
- Single-hop: possible to create a ciphertext which is not further transformable, via Enc'()
- In [LV08], Enc'() $\cong$ ReEnc(Enc())
  - a reason is that the ciphertext is re-randomizable
  - also explains why it is at most RCCA secure
- In our scheme, ReEnc() is deterministic
  - but Enc'() exists, also nontransformable
- Security definition for Enc'() is much simpler
  - usual CCA, Adv can get *all* re-encryption key
  - covers "master secret security" – recover *sk* in full

Token-Controlled Approach

- ReKeyGen selects a random token to hide (a form of) the delegator's secret
- This token is encrypted under the delegatee's public key, by a slightly different way
- Implicitly used in Shao-Cao 09 and 2 ID-based schemes (P.S. but not collusion resistant)

Our Attack (High Level)

- Re-encryption (not necessary of the challenge ciphertext) generates a cipherext which contains a part with partial information about the token
- No validity check of this part in decryption algorithm of Shao-Cao
- Possible fix requires a validity check, which means 1 more exponentiation

Overview of Our Scheme

- ElGamal encryption
  - with Fujisaki-Okamoto (FO) transformation and Schnorr signature for ciphertext integrity
- Re-encryption is done using a random token to hide the secret key
- Each user has 2 secret keys
  - Require both to decrypt an original ciphertext/ to create a transformation key
  - Encryption of random token in transformation key just requires one secret key to decrypt

KeyGen and Encryption

- $sk_i = (x_{i,1}, x_{i,2})$
- $(pk_{i,1}\ pk_{i,2}) = (g\text{^}(x_{i,1}), g\text{^}(x_{i,2}))$
- Let $pk_i = pk_{i,2} * pk_{i,1}\text{^}(H_4(pk_{i,2}))$
- FO: $r = H_1(m, w)$, w <- \$
- ElGamal: $E = pk^r$, $F = H_2(g^r) \oplus (m \,||\, w)$
- Schnorr: $D = (pk)^u$, $s = u + rH_3(D,E,F)$

Decryption

- $E = pk^r$, $F = H_2(g^r) \oplus (m \mid\mid w)$
- $D = (pk)^u$, $s = u + r * H_3(D, E, F)$
- Check if $pk^s = D * E$^$(H_3(D, E, F))$
- Define $sk = x_{i,1} H_4(pk_{i,2}) + x_{i,2}$
- $(m' \mid\mid w')$ <- $F \oplus H_2(E^{1/sk})$
- Return m' if $E = (pk)$^$(H_1(m', w'))$

ReKey and ReEnc (i -> j)

- Pick a random token h <- \$
- FO: $v = H_1(h, \pi)$, π <- \$
- ElGamal: $V = pk_{j,2}^{v}$, $W = H_2(g^{v}) \oplus (h || \pi)$
- $rk_{i \to j} = (h/sk_i, V, W)$
- ReEnc sees if $pk_i^{s} = D * E$^$(H_3(D,E,F))$
- Output ($E' = E$^$(h/sk_i) = g^{rh}$, F, V, W)

Enc' and Dec

- $E' = g^{rh}$, $F = H_2(g^r) \oplus (m \,||\, w)$
- $V = pk_{j,2}{}^{v}$, $W = H_2(g^v) \oplus (h \,||\, \pi)$
- *Enc*' (for nontransformable ctxt) picks h
- To decrypt, recover $(h \,||\, \pi)$, check it; recover g^r and hence $(m \,||\, w)$, check it

Intuition of Security

- rk has $h \,/\, (x_{i,1}\, H_4(pk_{i,2}) + x_{i,2})$
- Even with h, value of $x_{i,2}$ is unknown
  - "Token" in *rk* is protected by x_2
  - "Chain collusion" attack is not possible

Comparison

	Shao-Cao 09	Ours
Encrypt	$5t_{exp}$ (in $\mathbf{Z}_{N2}$)	$3t_{exp}$ (in $\mathbf{G}$)
ReEncrypt	$4t_{exp}$ (in $\mathbf{Z}_{N2}$)	$2.5t_{exp}$ (in $\mathbf{G}$)
Decrypt (Original)	$5t_{exp}$ (in $\mathbf{Z}_{N2}$)	$3.5t_{exp}$ (in $\mathbf{G}$)
Decrypt (Transformed)	$5t_{exp}$ (in $\mathbf{Z}_{N2}$)	$4t_{exp}$ (in $\mathbf{G}$)
Overhead (Original)	$3\|(N_X)^2\| + \|m\| + 2k$	$2\|\mathbf{G}\| + \|Z_q\| + k$
Overhead (Transformed)	$3\|(N_X)^2\| + 2\|(N_Y)^2\| + k$	$2\|\mathbf{G}\| + 2k$
Assumption	DDH over $\mathbf{Z}_{N2}$	CDH over $\mathbf{G}$
Remark	Decryption needs pk_X	N/A

Concluding Summary

- Unidirectional PRE schemes use pairings
  - Except Shao and Cao in PKC '09
- We showed that their CCA proof is flawed
- We present an efficient CCA-secure unidirectional PRE scheme without pairings
- Efficiency gain and CCA security may come from our (reasonable) weakening of the adversary model
  - "token" approach has been used implicitly
  - but the model was never adjusted to match

Summary of Summary

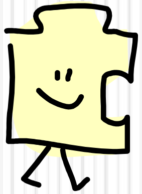
- Model
- Attack
- Construction
  - Better efficiency (albeit the proof assumes random oracle)
  - More standard complexity assumption

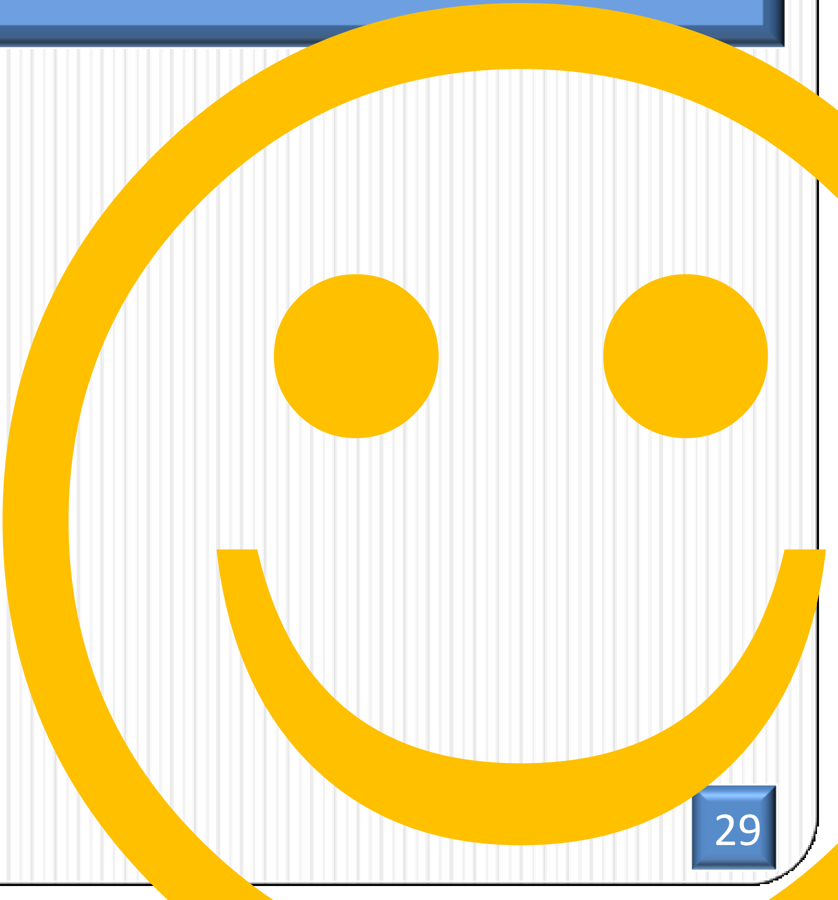
Open Problems

- Pairing-free CCA-secure scheme with no weakening of security model
- Proxy re-cryptography without pairing
  - conditional proxy re-encryption
  - proxy re-signatures, etc

Thank you very much!

- Questions/comments are welcome.
- schow@cs.nyu.edu





Collusion Attack of Shao-Cao

- A collusion of a delegatee of X (say Y) and his proxy can recover a weak secret key of X, wsk_X
- Re-encrypting X's ciphertext to *other* delegatee retains *most* part of the original one
- In particular, it is decryptable by wsk_X
- Z is the target, X is the delegator, and compromise Y and the proxy of X for Y