On the Impossibility of Batch Update for Cryptographic Accumulators

Philippe Camacho (pcamacho@dcc.uchile.cl)
Alejandro Hevia (ahevia@dcc.uchile.cl)

FMCrypto Workshop: Formal Methods in Cryptography
University of Chile

March 24, 2010

Introduction

This work is about an impossibility result...

- [FN02]
  Open problem:
  “Can we build accumulators with Batch Update?”
- [WWP07, WWP08]
  - Construction for accumulators with Batch Update.
  - Problem: the construction is **not secure**.
  - 8 papers(without ours) cite [WWP07], two of them build upon [WWP07].
- [CH09](our work): **Batch Update is impossible!**

Notion of Cryptographic Accumulator

- Problem
    - A set X
    - Given an element x: prove/verify $x \in X$
- Let $X = \{x_1, ..., x_n\}$
    - X will be represented by a short value Acc_X
    - Verify(x, w, Acc_X): returns Yes whether $x \in X$
- Vocabulary
    - Acc_X is called the *accumulated value* for X
    - w is called a witness

Participants

- Manager
  - Computes setup values
  - Computes the accumulated value Acc
  - Computes the witness w_x for a given x
- User
  - Ask for element insertion or deletion to the Manager
  - Ask for witness computation to the Manager
  - Check whether $x \in X$ using Acc, w_x and x

Applications

- Time-stamping [BdM94]
- Anonymous Credentials [CL02]
- Broadcast Encryption [GR04]
- Certificate Revocation List [LLX07]
- ....

Some properties

- Dynamic / Static
- Weak / Strong
- Universal (non-membership proofs)

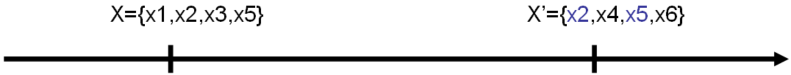
In our case we study dynamic accumulators that are **dynamic**, **not strong** and **not universal**.

Operations (1/2)

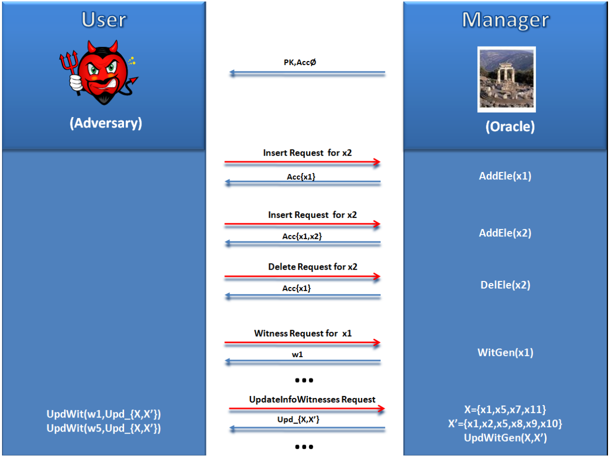
Algorithm	**Returns**	**Run by**
$\mathrm{KeyGen}(1^k)$	$(PK, SK), Acc_{\emptyset}$	*Manager*
$\mathrm{AddEle}(x, Acc_X, SK)$	$Acc_{X \cup \{x\}}$	*Manager*
$\mathrm{DelEle}(x, Acc_X, SK)$	$Acc_{X \setminus \{x\}}$	*Manager*
$\mathrm{WitGen}(x, Acc_X, SK)$	witness w for x relative to Acc_X	*Manager*
$\mathrm{Verify}(x, w, Acc_X, PK)$	returns Yes whether $x \in X$	*User*

Operations (2/2)

Algorithm	**Returns**	**Run by**
UpdWitGen(X, X', SK)	$Upd_{X,X'}$ for the elements $x \in X \cap X'$.	*Manager*
UpdWit$(w, Upd_{X,X'}, PK)$	new witness w' for $x \in X'$	*User*



Security Model ([CL02])



User

(Adversary)

Manager

(Oracle)

PK,AccØ

Insert Request for x2

Acc{x1}

AddEle(x1)

Insert Request for x2

Acc{x1,x2}

AddEle(x2)

Delete Request for x2

Acc{x1}

DelEle(x2)

Witness Request for x1

w1

WitGen(x1)

•••

UpdateInfoWitnesses Request

Upd_{X,X'}

X={x1,x5,x7,x11}
X'={x1,x2,x5,x8,x9,x10}
UpdWitGen(X,X')

UpdWit(w1,Upd_{X,X'})
UpdWit(w5,Upd_{X,X'})

•••

$$\Pr\left[\text{Verify}(x, w, Acc_{X'}, PK) = \text{Yes} \wedge x \notin X'\right] = neg(k)$$

The Batch Update Property ([FN02])

Definition

(*Batch Update for accumulator schemes*). Let $\mathfrak{Acc}$ be an accumulator scheme. $\mathfrak{Acc}$ has the *Batch Update* property if for every pair (X, X') we have $|Upd_{X,X'}| = O(k)$ where k is the security parameter.

In other words, the information needed to update all the user's witnesses should have size independent w.r.t the cardinality of the sets X, X'.

Problem with the construction of [WWP07]

Description of the Attack

- $X_0 = \emptyset$
- Insert x_1. $X_1 = \{x_1\}$
- Delete x_1. $X_2 = \emptyset$
- Ask for the update information Upd_{X_1,X_2}
- With Upd_{X_1,X_2} I can update my witness w_{x_1}
- **But** $x_1 \notin X_2$**!**

Our result

Theorem

For an update involving m delete operations in a set of N elements, the size of the information $Upd_{X,X'}$ required by the algorithm UpdWit *while keeping the dynamic accumulator secure is $\Omega(m \log \frac{N}{m})$. In particular if $m = \frac{N}{2}$ with N even, we have $|Upd_{X,X'}| = \Omega(m)$.*

Corollary

Cryptographic accumulators with Batch Update do not exist.

Proof of Corollary.

$|X| = p(k)$ where p is a polynomial.
Then $|Upd_{X,X'}| = \Omega(|X|) = \Omega(p(k)) = \omega(k)$.

□

Proof of the Theorem

Proof.

- $X = \{x_1, \ldots, x_N\}$
- The *Manager* deletes m elements from X
- New set $X' = X \setminus X_d$ where $X_d = \{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$
- The *Manager* sends $Upd_{X,X'}$ to the *User*
- The user runs UpdWit on every witness w_x for $x \in X$
  - $w'_x = \mathrm{UpdWit}(w_x, Upd_{X,X'}, PK)$ is valid $\Rightarrow x \in X'$ else $x \notin X'$
- So only with the information contained in $Upd_{X,X'}$ the *User* can rebuild X_d
- How much information is needed to code X_d?
  - $log(\binom{N}{m})$
  - $\binom{N}{m} \geq (\frac{N}{m})^m$
  - $|Upd_{X,X'}| \geq m \log \frac{N}{m}$

□

Thank you!

`http://www.dcc.uchile.cl/~pcamacho`

Josh C Benaloh and Michael de Mare.
One-Way Accumulators: A Decentralized Alternative to Digital Signatures.
Lecture Notes in Computer Science, 765:274—-??, 1994.

Philippe Camacho and Alejandro Hevia.
On the impossiblity of batch update for cryptographic accumulators.
Technical report, 2009.

Jan Camenisch and Anna Lysyanskaya.
Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials.
Lecture Notes In Computer Science; Vol. 2442, 2002.

Nelly Fazio and Antonio Nicolisi.
Cryptographic Accumulators: Definitions, Constructions and Applications.
Technical report, 2002.

Craig Gentry and Zulfikar Ramzan.
RSA Accumulator Based Broadcast Encryption.
In *Information Security*, pages 73–86. 2004.

Jiangtao Li, Ninghui Li, and Rui Xue.
Universal Accumulators with Efficient Nonmembership Proofs.
In *Applied Cryptography and Network Security*, pages 253–269. 2007.

Peishun Wang, Huaxiong Wang, and Josef Pieprzyk.
A New Dynamic Accumulator for Batch Updates.
In *Information and Communications Security*, pages 98–112. 2007.

Peishun Wang, Huaxiong Wang, and Josef Pieprzyk.
Improvement of a Dynamic Accumulator at ICICS 07 and Its Application in Multi-user Keyword-Based Retrieval on Encrypted Data.

In *Asia-Pacific Conference on Services Computing. 2006 IEEE*, volume 0, pages 1381–1386, Washington, DC, USA, 2008. IEEE Computer Society.