The Bilinear Pairing-based Accumulator Proposed at CT-RSA'05 is not Collision Resistant

Christophe Tartary[1] and Huaxiong Wang[1,2]

[1] Centre for Advanced Computing, Algorithms and Cryptography
Department of Computing
Macquarie University
NSW 2109 Australia
[2] Division of Mathematical Sciences
School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore
{ctartary,hwang}@ics.mq.edu.au

Abstract. In this paper, we demonstrate that the construction proposed by Lan Nguyen at CT-RSA'05 does lead to a cryptographic accumulator which is not collision resistant.

Keywords: bilinear pairing, collision resistance, cryptographic accumulators.

1 Introduction

A cryptographic accumulator is an algorithm allowing the aggregation of a large set of elements into a single value of constant size. Accumulators were introduced by Belanoh and de Mare [2] in order to design distributed protocols without the presence of a trusted central authority. Such constructions are used in time-stamping [2], fail-stop signatures [1], ring signatures [4] and multicast stream authentication [5] for instance. Camenisch and Lysyanskaya introduced the notion of dynamic accumulators which allow the addition and deletion of values from the original set of elements [3]. In 2005, Nguyen proposed a dynamic accumulator based on bilinear pairings to design ID-based ad-hoc anonymous identification schemes and identity escrow protocols with membership revocation.

In this article we demonstrate that the accumulator suggested by Nguyen is not collision resistant which constitutes a main weakness for the different constructions relying on its security.

The rest of this paper is organized as follows. In the next section, we will recall the definitions and results from the original paper by Nguyen [7]. In Sect. 3, we will design our attack against the collision resistance of Nguyen's accumulator.

2 Preliminaries

In this section, we recall the definitions and constructions as they appear in Nguyen's article [7].

2.1 Notations and Terminology

Definition 1. *A function* $f : \mathbb{N} \to \mathbb{R}^+$ *is said to be* negligible *if:*

$$\forall \alpha > 0\, \exists \ell_0 \in \mathbb{N} : \forall \ell > \ell_0 \quad f(\ell) < \ell^{-\alpha}$$

Definition 2. *A function* $f : \mathbb{N} \to \mathbb{R}^+$ *is said to be* polynomially bounded *if:*

$$\exists \alpha_0 > 0 : \forall \ell \in \mathbb{N} \quad f(\ell) < \ell^{\alpha_0}$$

We denote $\mathbb{Z}_\mathrm{p}$ the set of residues $\{0, \dots, p-1\}$ modulo p. We consider two additive cyclic groups $\mathbb{G}_1 =< P_1 >$ and $\mathbb{G}_2 =< P_2 >$ as well as a cyclic multiplicative group $\mathbb{G}_\mathrm{M}$. These three groups are assumed to have the same prime order p. We assume that we have a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\mathrm{M}$ such that:
1. $\forall (P, Q) \in \mathbb{G}_1 \times \mathbb{G}_2\ \forall (a, b) \in \mathbb{Z}_\mathrm{p} \times \mathbb{Z}_\mathrm{p} \quad e(a\,P, b\,Q) = e(P, Q)^{a\,b}$
2. $e(\cdot, \cdot)$ is not degenerated: $e(P_1, P_2) \neq 1$
3. There exists a computationally efficient algorithm to compute $e(P, Q)$ for every couple (P, Q) from $\mathbb{G}_1 \times \mathbb{G}_2$.

As in [7], we consider $\mathbb{G}_1 = \mathbb{G}_2$ (and thus $P_1 = P_2$) in the remaining of this article. We have the following definition:

Definition 3. *A* bilinear pairing instance generator *is a probabilistic polynomial-time* (PPT) *algorithm* $\mathcal{G}$ *taking as input a security parameter* 1^ℓ *and returning a uniformly random tuple* $\mathbf{t} = (p, \mathbb{G}_1, \mathbb{G}_M, e(\cdot, \cdot), P)$ *of bilinear pairing parameters defined as before where* ℓ *represents the length of the prime number* p *and* $\mathbb{G}_1 =< P >$.

We now present the definition of accumulators and the collision resistance property as set by Nguyen in [7].

Definition 4. *An* accumulator *is a tuple* $(\{\mathbf{X}_\ell\}_{\ell \in \mathbb{N}}, \{\mathbf{F}_\ell\}_{\ell \in \mathbb{N}})$, *where* $\{\mathbf{X}_\ell\}_{\ell \in \mathbb{N}}$ *is called the* value domain *of the accumulator and* $\{\mathbf{F}_\ell\}_{\ell \in \mathbb{N}}$ *is a sequence of pairs of functions such that each* $(f, g) \in \mathbf{F}_\ell$ *is defined as* $f : \mathbf{U}_f \times \mathbf{X}_f^{ext} \to \mathbf{U}_f$ *for some* $\mathbf{X}_f^{ext} \supset \mathbf{X}_\ell$ *and* $g : \mathbf{U}_f \to \mathbf{U}_g$ *is a bijective function. In addition the following properties are satisfied:*

(Efficient Generation) *There exists an efficient algorithm* $\mathcal{G}$ *taking as input a security parameter* 1^ℓ *and outputting a random element* (f, g) *from* $\mathbf{F}_\ell$ *possibly together with some auxiliary information* a_f.

(Quasi-commutativity) $\forall \ell \in \mathbb{N}\ \forall (f, g) \in \mathbf{F}_\ell\ \forall u \in \mathbf{U}_f\ \forall (x_1, x_2) \in \mathbf{X}_\ell \times \mathbf{X}_\ell$ $f(f(u, x_1), x_2) = f(f(u, x_2), x_1)$. *For any* $\ell \in \mathbb{N}, (f, g) \in \mathbf{F}_\ell$ *and* $\mathbf{X} := \{x_1, \dots, x_q\} \subset \mathbf{X}_\ell$, *we call* $g(\cdots f(u, x_1) \cdots, x_q)$ *the* accumulated value *of the set* $\mathbf{X}$ *over* u. *It does not depend on the order of the elements to be evaluated and is denoted* $f(u, \mathbf{X})$.

(Efficient Evaluation) *For any* $(f, g) \in \mathbf{F}_\ell, u \in \mathbf{U}_f$ *and* $\mathbf{X} \subset \mathbf{X}_\ell$ *with polynomially bounded size (as a function of* ℓ*),* $g(f(u, \mathbf{X}))$ *is computable in time polynomial in* ℓ *even without the knowledge of* a_f.

Nguyen set the previous definition to generalize the accumulator constructions by Camenisch and Lysyanskaya [3] and Dodis et al. [4] where $\mathbf{U}_f = \mathbf{U}_g$ and the bijective function g is the identity function.

Definition 5 (Collision Resistant Accumulator). *An accumulator is said to be* collision resistant *if for every PPT algorithm $\mathcal{A}$, the function:*

$$\mathrm{Adv}_{\mathcal{A}}^{\textit{col.acc}}(\ell) := \mathrm{Prob}\Big((f,g) \stackrel{R}{\leftarrow} \mathbf{F}_\ell; u \stackrel{R}{\leftarrow} \mathbf{U}_f; (x, w, \mathbf{X}) \leftarrow \mathcal{A}(f, g, \mathbf{U}_f, u) \mid (\mathbf{X} \subset \mathbf{X}_\ell) \wedge (w \in \mathbf{U}_g) \wedge (x \in \mathbf{X}_f^{\mathrm{ext}} \setminus \mathbf{X}) \wedge (f(g^{-1}(w), x) = f(u, \mathbf{X}))\Big)$$

is negligible as a function of ℓ. We say that w is a witness *for the fact that $x \in \mathbf{X}_\ell$ has been accumulated in $v \in \mathbf{U}_g$ whenever $g(f(g^{-1}(w), x)) = v$.*

We now introduce the $q-$Strong Diffie Hellman ($q-$SDH) assumption as it was used by Nguyen to prove the security of his construction.

Definition 6. *The $q-$SDH assumption states that for every PPT algorithm $\mathcal{A}$, the function:*

$$\mathrm{Adv}_{\mathcal{A}}^{\textit{q-SDH}}(\ell) := \mathrm{Prob}\left(\left(\mathcal{A}(\mathbf{t}, P, s\,P, \ldots, s^{\mathrm{q}}\,P) = \left(c, \tfrac{1}{s+c}\,P\right)\right) \wedge (c \in \mathbb{Z}_p)\right)$$

is negligible as a function of ℓ where $\mathbf{t} = (p, \mathbb{G}_1, \mathbb{G}_M, e(\cdot,\cdot), P) \leftarrow \mathcal{G}(1^\ell)$ and $s \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$.

2.2 Construction of the Accumulator

To generate an instance of the accumulator from the security parameter ℓ, we run the algorithm $\mathcal{G}$ on input 1^ℓ to obtain a tuple $\mathbf{t}$ and a uniformly chosen element s from $\mathbb{Z}_{\mathrm{p}}^*$ as in Definition 6. We construct a tuple $\mathbf{t}' := (P, s\,P, \ldots, s^{\mathrm{q}}\,P)$ where q is an upper bound on the number of elements to be accumulated. The corresponding functions (f, g) for this instance $(\mathbf{t}, \mathbf{t}')$ are defined as:

$$\begin{array}{rcl} f : \mathbb{Z}_{\mathrm{p}} \times \mathbb{Z}_{\mathrm{p}} & \longrightarrow & \mathbb{Z}_{\mathrm{p}} \\ (u, x) & \longmapsto & (x+s)\,u \end{array} \qquad \begin{array}{rcl} g : \mathbb{Z}_{\mathrm{p}} & \longrightarrow & \mathbb{G}_1 \\ u & \longmapsto & u\,P \end{array}$$

This construction involves that we have: we have:

$$\mathbf{U}_f = \mathbf{X}_f^{\mathrm{ext}} = \mathbb{Z}_{\mathrm{p}} \qquad \mathbf{U}_g = \mathbb{G}_1 \qquad \mathbf{X}_\ell = \mathbb{Z}_{\mathrm{p}} \setminus \{-s\}$$

It is clear that f is quasi-commutative. In addition for $u \in \mathbb{Z}_{\mathrm{p}}$ and a set $\boldsymbol{X} = \{x_1, \ldots, x_k\} \subset \mathbb{Z}_{\mathrm{p}} \setminus \{-s\}$ where $k \leq q$, the accumulated value $g(f(u, \boldsymbol{X})) = \left(\prod_{i=1}^{k} (x_i + s)\,u\right) P$ is computable in time polynomial in ℓ from the tuple $\boldsymbol{t}'$ and without the knowledge of the auxiliary information s [7].

We now recall the security theorem demonstrated by Nguyen:

Theorem 1 ([7]). *The accumulator related to the pair (f, g) defined above provides collision resistance if the $q-$SDH assumption holds, where q is the upper bound on the number of elements to be accumulated.*

3 Breaking the Collision Resistance

In this section, we construct a PPT algorithm $\mathcal{A}$ which breaks the collision resistance property of the accumulator with non-negligible probability. Since this will contradict the result from Theorem 1, we will then show that the adversary reduction model to the $q-$SDH assumption given by Nguyen was incorrect.

3.1 Our Attack

Algorithm Construction. According to Definition 5, the adversary is given the functions f and g as well as u and the set $\mathbf{U}_f = \mathbb{Z}_\mathrm{p}$. We build the following algorithm:

Algorithm $\mathcal{A}$
Input: The pair of functions (f, g) and the value u.
1. Compute $s = f(1, 0)$
2. Let k be any polynomial function of ℓ. Choose uniformly at random $k+1$ elements of $\mathbb{Z}_\mathrm{p} \setminus \{-s\}$ denoted $x_1, \ldots, x_k, x$ and set $\mathbf{X} := \{x_1, \ldots, x_k\}$.
3. Compute $\lambda := \prod_{i=1}^{k}(x_i + s)\, u \bmod p$ and $\mu := (x+s)^{-1} \bmod p$. Denote $\xi := \lambda\, \mu \bmod p$ and set $w := g(\xi)$.
Output: The triple (x, w, X).

Correctness of the output. Due to Step 2, we have: $\mathbf{X} \subset \mathbf{X}_\ell$ and $x \in \mathbf{X}_f^{\mathrm{ext}} \setminus \mathbf{X}$. From Step 3, we obtain: $w \in \mathbf{U}_g$.

By construction of $\mathbf{X}$ we have: $f(u, \mathbf{X}) = \prod_{i=1}^{k}(x_i + s)\, u \bmod p$. We also have $\xi = g^{-1}(w)$ since g is invertible. We obtain the following equalities:

$$\begin{aligned} f(\xi, x) &= (x+s)\, \xi \bmod p \\ &= (x+s)\, \lambda\, \mu \bmod p \\ &= (x+s)\, (x+s)^{-1}\, \lambda \bmod p \\ &= \lambda \bmod p \\ &= \lambda \\ &= f(u, \mathbf{X}) \end{aligned}$$

Therefore we have: $f(g^{-1}(w), x) = f(u, \mathbf{X})$. In addition the construction of the triple $(x, w, \mathbf{X})$ is deterministic (the value μ always exists since $x \neq -s$). So we obtain:

$$\mathrm{Adv}_{\mathcal{A}}^{\text{col.acc}}(\ell) = 1$$

Running time. First it should be noticed that any operation (addition, multiplication, inversion) in $\mathbb{Z}_\mathrm{p}$ can be done in quadratic time as a function of ℓ [6]. That is, any of these arithmetic operations can be performed in $O(\ell^2)$ bit operations.

Since k is a polynomial function of ℓ, we denote it as $\mathcal{K}(\ell)$. We can also assume that picking one random element from $\mathbb{Z}_{\rm p} \setminus \{-s\}$ requires polynomial time $\mathcal{R}(\ell)$ (otherwise it would be computationally infeasible to construct a single family of elements from $\mathbb{Z}_{\rm p} \setminus \{-s\} = \mathbf{X}_\ell$ which is not a realistic assumption). Thus Step 2 is executed in $(\mathcal{K}(\ell)+1)\,\mathcal{R}(\ell)$ bit operations.

Since s has been obtained at Step 1 (using $O(\ell^2)$ bit operations), one can get λ with k multiplications and k additions in $\mathbb{Z}_{\rm p}$ representing $O(\mathcal{K}(\ell)\,\ell^2)$ bit operations. Each of the two elements, μ and ξ, also needs $O(\ell^2)$ bit operations to be computed while g can be run in polynomial time $\mathcal{G}(\ell)$. Therefore the number of bit operations executed during Step 3 is $O(\mathcal{K}(\ell)\,\ell^2 + \mathcal{G}(\ell))$.

As a consequence, the running time of $\mathcal{A}$ is:

$$O(\ell^2) + (\mathcal{K}(\ell)+1)\,\mathcal{R}(\ell) + O(\mathcal{K}(\ell)\,\ell^2 + \mathcal{G}(\ell)) = O(\mathcal{K}(\ell)\,\mathcal{R}(\ell)\,\ell^2 + \mathcal{G}(\ell))$$

which is polynomial in the security parameter ℓ.

Therefore $\mathcal{A}$ is a PPT algorithm breaking the collision-resistance of the accumulator with non-negligible probability. Thus the accumulator is not collision-resistant. We point out that $\mathcal{A}$ enables to construct many such triples $(x, w, \mathbf{X})$.

3.2 Comments on the Original Security Proof

The proof of Theorem 1 given by Nguyen in [7] might be right but the adversary reduction is not accurate. According from Definition 6, an enemy trying to break the $q-$SDH assumption should only be provided with $(\mathbf{t}, P, z\,P, \ldots, z^q\,P)$. Nevertheless the adversary model of the accumulator allows the enemy to query f and g. As a consequence, it is easy for him to obtain z by a single query to f as in Step 1 of $\mathcal{A}$. Then he can compute $(z+c)^{-1}$ mod p in $O(\ell^2)$ bit operations for *any* c. Finally he runs g on that inverse and obtain $\frac{1}{z+c}\,P$. This means that the $q-$SDH assumption is *never* verified in Nguyen's enemy model. Thus the security benefit of Theorem 1 vanishes.

In order to be immune against our attack, Nguyen suggested to allow the adversary the use of the composition $g \circ f$ instead of both f and g [8]. His new definition is as follows:

Definition 7. *An accumulator is said to be* collision resistant *if for every PPT algorithm $\mathcal{A}$, the function:*

$$\text{Adv}_{\mathcal{A}}^{\textit{col.acc}}(\ell) := \text{Prob}\Big((f,g) \overset{R}{\leftarrow} \mathbf{F}_\ell; u \overset{R}{\leftarrow} \mathbf{U}_f; (x,w,\mathbf{X}) \leftarrow \mathcal{A}(g\circ f, \mathbf{U}_f, u)\,|\, (\mathbf{X} \subset \mathbf{X}_\ell) \wedge (w \in \mathbf{U}_g) \wedge (x \in \mathbf{X}_f^{\text{ext}} \setminus \mathbf{X}) \wedge (f(g^{-1}(w), x) = f(u, \mathbf{X}))\Big)$$

is negligible as a function of ℓ. We say that w is a witness *for the fact that $x \in \mathbf{X}_\ell$ has been accumulated in $v \in \mathbf{U}_g$ whenever $g(f(g^{-1}(w), x)) = v$.*

One can notice that the enemy is still allowed access to g since u is given. The accuracy of this new definition for collision resistance remains to be justified. In order to apply Theorem 1 it must be demonstrated that the view of an adversary wishing to break the collision resistance of the accumulator can be reduced to the view of someone trying to break the $q-$SDH assumption. In particular, it must be argued that given $g \circ f, \mathbf{U}_f, u$ and the public parameters $(\mathbf{t}, \mathbf{t}')$, the adversary cannot get the secret value s in polynomial time with non-negligible probability (otherwise he can perform the same attack as in Sect. 3.1).

4 Conclusion

In this paper, we showed that the construction from [7] did not give a collision resistant accumulator. As a consequence, the security of the identity escrow protocol and the ID-based identification scheme developed in [7] is not guaranteed any longer. The reader may be aware that Zhang and Chen already exhibited problems in the ID-based identification protocol [9]. Nevertheless they did not notice that the accumulator could be directly attacked.

Acknowledgment

The authors would like to thank Doctor Lan Nguyen for his comments related to the security of his construction. This work was supported by the Australian Research Council under ARC Discovery Projects DP0558773 and DP0665035. The first author's work was also funded by an iMURS scholarship supported by Macquarie University.

References

[1] Niko Barić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology - Eurocrypt'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 480 – 494, Konstanz - Germany, May 1997. Springer - Verlag.

[2] Josh Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital signatures. In *Advances in Cryptology - Eurocrypt'93*, volume 765 of *Lecture Notes in Computer Science*, pages 274 – 285, Lofthus, Norway, May 1993. Springer - Verlag.

[3] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology - Crypto'02*, volume 2442 of *Lecture Notes in Computer Science*, pages 61 – 76, Santa Barbara, USA, August 2002. Springer - Verlag.

[4] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In *Advances in Cryptology - Eurocrypt'04*, volume 3027 of *Lecture Notes in Computer Science*, pages 609 – 626, Interlaken, Switzerland, May 2004. Springer.

[5] Chris Karlof, Naveen Sastry, Yaping Li, Adrian Perrig, and J. D. Tygar. Distillation codes and applications to DoS resistant multicast authentication. In *11th Network and Distributed Systems Security Symposium (NDSS)*, San Diego, USA, February 2004.

[6] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[7] Lan Nguyen. Accumulators from bilinear pairings and applications. In *Topics in Cryptology CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 275 – 292, San Francisco, USA, February 2005. Springer - Verlag.

[8] Lan Nguyen. Private communication, November 2006.

[9] Fangguo Zhang and Xiaofeng Chen. Cryptanalysis and improvement of an ID-based ad-hoc anonymous identification scheme at CT-RSA 05. Available online at: http://eprint.iacr.org/2005/103.pdf, March 2005.