Efficient Batch Verification of Signature Schemes based on Bilinear Maps

Noel McCullagh
noel.mccullagh@computing.dcu.ie

School of Computing,
Dublin City University,
Glasnevin
Dublin 9.
Ireland.**

Abstract. In this paper we present batch signature verification schemes for identity and non-identity signatures schemes based on bilinear maps. We examine some signature schemes and exploit their properties so that we can batch process the verification of these signatures in an efficient manner. Batch verification of message signatures is useful in real world applications. Most email clients are predominantly offline and so do not download emails one at a time. Instead the mails arrive at an online mail server individually, where they are collected together and stored. It is only after some period of time that any mails on the server are downloaded in bulk. It is not unreasonable to have 5 - 10 emails download into your inbox in any one transaction with the mail server. Say these mails were all signed, then this would be an ideal time to do batch signature verification. We show that we can make substantial savings over the naïve approach of verifying one message signature at a time.

Keywords: signature, identity based, batch verification

1 Background

Online personal messaging systems are one of the oldest and most popular internet applications. It is predicted that there will be in excess of 36 billion emails sent every day in 2005 [11]. This number is added to by SMS "txt messaging" and other services. A large amount of these messages are downloaded in bulk by the end user, as they are not online all of the time. They may have to make an expensive telephone call to dial-up to their ISP's server or they may have their mobile phone turned off when the SMS message is sent. Whatever the reason, it is common to download more than one message in a single transaction with the mail server.

** The author wishes to thank Enterprise Ireland for their support with this research under grant IF/2002/0312/N.

Traditionally for pairing based signature schemes verification is substantially more computationally expensive than signing *(verification requires pairing operations - signing does not)*. This bulk downloading gives us an opportunity to perform batch signature verification, and so somewhat lessen the impact of this computationally expensive process.

Of course, batch processing is not a new idea. There have been several papers published on the subject, for example [7,5]. Batch verification of signatures has attracted particular attention since signatures are generated once and may be verified often. There are also particular instances in the "real world" where many signatures have to be checked sequentially. Examples include checking Cerificate Authority certificate chains or checking batch transactions such as the bulk downloading of email mentioned above [1,8,13]. Until now, there has been no research into the efficient batch verification of signatures based on bilinear maps.

We suggest batch verification schemes to work with both identity based and non-identity based signature schemes. In an identity based cryptosystem the public key can be any arbitrary value. For convenience we make the public key the online identity or *identifier* of the entity. The identifier can be anything that uniquely identifies an entity in a given context. This is convenient, as every online computer already has a unique identifier as part of IPv4 and IPv6. Everyone with email has a unique address and every phone number in the world is unique. Therefore to send an encrypted mail we only have to know the persons email address, to check an identity based signature we only have to know the email address that the message was sent from.

2 Mathematical Preliminaries

An elliptic curve $\mathbb{E}(\mathbb{F}_{q^k})$ is the set of solutions (x, y) over the field $\mathbb{F}_{q^k}$ to an equation of the form $y^2 = x^3 + Ax + B$, together with an additional point at infinity, denoted O. There exists an abelian group law on $\mathbb{E}$. There are explicit formulas for computing the coordinates of a point $P_3 = P_1 + P_2$ from the coordinates of P_1 and P_2. Scalar multiplication of a point is defined as the repeated addition of a point to itself n times, e.g. $3P_1 = P_1 + P_1 + P_1$.

The number of points of an elliptic curve $\#\mathbb{E}(\mathbb{F}_{q^k})$ is called the order of the curve over the field $\mathbb{F}_{q^k}$. A point P has order r if $rP = O$ for the smallest possible positive integer value of r. The set of all points of order r in $\mathbb{E}$ is denoted $\mathbb{E}[r]$. This is arranged as cyclic subgroups of prime order r. The order of a point always evenly divides the curve order. A subgroup $\mathbb{G}$ of an elliptic curve is said to have embedding degree *(a.k.a security multiplier)* k if its order r divides $q^k - 1$ for the smallest possible integer value of k. We assume $k > 1$. [9]

2.1 Properties of Bilinear Maps

Our batch verification system can be implemented using any pairing algorithm *(Tate, Weil, modified Weil or modified Tate)* on which the original signature

schemes can be based. While we realise that the Tate pairing is preferable to the Weil pairing in terms of performance we will use the Weil pairing in this paper for clarity of exposition. The modified Weil pairing $\hat{e}(P,Q)$ is $e(P,\Phi(Q))$ where $e(.,.)$ is the Weil pairing and $\Phi(.)$ is an efficiently computable group automorphism [12,9]. The modified Weil pairing is an example of a bilinear map of the form e: $\mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$ where $\mathbb{G}_0$ and $\mathbb{G}_1$ are groups of order r. We will assume thoughout the rest of this paper that we are using the modified Weil pairing.

- The modified Weil pairing of elements of the group $\mathbb{G}_0$
  $\hat{e}(P,Q) = \hat{e}(Q,P)$

- Bilinearity of the Pairing
  $\hat{e}(xP,yQ) = \hat{e}(P,Q)^{xy}$

3 The Proposed Signature Scheme

We propose an efficient "batch short signature scheme" that incoporates elements of the BLS short signature scheme [2] and Zhang *et al's* short signature scheme [15]. We also propose an efficient batch verification process for Cha & Cheon's [4] and Yi's [14] Identity based signature schemes. First we will recap briefly on these schemes in order to refamiliarise the reader with them. We do not go into detail here and suggest the interested reader consult the original papers for a full description and proof of security.

3.1 The BLS Short Signature Scheme

It is assumed that there exists a hash function $h(.) : \{0,1\}^* \rightarrow \{\mathbb{G}_1 \backslash O\}$ which maps arbitrary bit strings onto elements of the group $\mathbb{G}_1$, but not to the identity element *(the point at infinity)*. This hash function should have the same properties as that described by Boneh and Franklin to map identities to points in their paper on identity based encryption [3].

Setup: The entity generates a random secret key $x \in \mathbb{Z}_r^*$. The public key is calculated as xP, for some publicly known group generator point P of the group $\mathbb{G}_0$. Obviously calculating the private key from the public key is the Elliptic Curve Discrete Logarithm Problem. This is believed to be hard in groups appropriate for cryptographic use. The public key is made public and its legitimacy assured by a top level certificate authority.

Signature: The message is hashed to a point on the curve, M, using the function *hash* described above. The signature is calculated as xM.

Verification: The BLS signature is verified using a bilinear map as follows:

$\hat{e}(public\ key, h(message)) \stackrel{?}{=} \hat{e}(group\ generator\ point,\ signature)$

To see that this is so, consider the following:

$\hat{e}(public\ key,\ h(message)) =$
$\hat{e}(xP, M) =$
$\hat{e}(P, M)^x =$
$\hat{e}(P, xM) =$
$\hat{e}(group\ generator\ point,\ signature)$

3.2 The Zhang *et al* Short Signature Scheme

Setup: In the Zhang *et al* short signature scheme the entity again generates a random secret key $x \in \mathbb{Z}_r^*$.The corresponding public key is xP, again P is a publicly known generator of the group $\mathbb{G}_0$.

Signature: The message is hashed using the hash function $hash(.) : \{0,1\}^* \rightarrow \mathbb{Z}_r^*$. The signature on the message is

- $h = hash(message)$
- $S = (x + h)^{-1}P$

Verification: The signature is now verified as follows:

- $h = hash(message)$
- $\hat{e}(S, hP + public\ key) \stackrel{?}{=} \hat{e}(P, P)$

To see that this is so:

$\hat{e}(S, hP + public\ key) =$
$\hat{e}(S, hP + xP) =$
$\hat{e}((h + x)^{-1}P, (h + x)P) =$
$\hat{e}(P, P)$

3.3 The new Batch Short Signature Scheme

We alter this signature scheme very slightly. Instead of calculating the public key as xP as described above, we calculate the public key as $x^{-1}P$. The message is signed as xM. The reason of this alteration will be explained in the next section.

The signature is verified iff the following equality holds:
$\hat{e}(public\ key,\ signature) \stackrel{?}{=} \hat{e}(group\ generator\ point,\ h(message))$

To see that this is so, consider the following:

$\hat{e}(public\ key,\ signature) =$
$\hat{e}(x^{-1}P, xM) =$
$\hat{e}(P, M)^{xx^{-1}} =$
$\hat{e}(P, M) =$
$\hat{e}(group\ generator\ point,\ h(message))$

Proof of Security of the Batch Short Signature Scheme

The proof of security of the Batch Short Signature Scheme is surprisingly simple. It is the definition of a believed to be hard cryptographic problem known as the "Bilinear Pairing Inversion Problem" (BPI). A reduction from this problem to the Discrete Logarithm Problem in the group $\mathbb{G}_2$ is shown by Yacobi in [12].

The bilinear pairing inversion problem, as defined by Yacobi is:

BPI: Given $P \in \mathbb{G}_0, \hat{e}(P, Q) \in \mathbb{G}_2$ find Q.

Rewriting this problem using our own notation we have:

BPI: Given $x^{-1}P \in \mathbb{G}_0, \hat{e}(P, M) \in \mathbb{G}_2$ find xM.

So forging a signature is as hard as solving the BPI problem.

3.4 Cha & Cheon's Identity Based Signature Scheme

Bilinear maps such as the Weil and Tate pairing allow us to quickly solve the Decision Diffie Hellman problem in appropriate groups. The Cha & Cheon identity based signature [4] essentially constructs two points and we can use Decision Diffie Hellman to establish if one point is a particular multiple of the other, even if we do not know the exact value of the multiplier. In common with most other identity based system both these schemes use the services of a Key Generation Centre (KGC).

Setup: The KGC generates a random master secret key $s \in \mathbb{Z}_r^*$. The KGC publishes the points P and sP, P is a group generator element of the group $\mathbb{G}_0$. Again, calculating the private key from the public key is the Elliptic Curve Discrete Logarithm Problem.

Extract: The KGC authenticates a user for a particular online identifier (i.e. checks that that user has a valid claim to that online identity). Once this has been established the private key is generated as sID where the users identity hashes to the point ID under some publicly known hash algorithm. Let's say for convenience in notation that Alices identifier "alice@company.com" maps to the point A, and her private key is sA.

Signature: Alice generates a random challenge R. She hashes the message and R to obtain $h \in \mathbb{Z}_r^*$. Alice signs the message by producing the value S below. The signature is the pair of points R and S.

- $R = rA$
- $h = hash(message, R)$
- $S = (r + h)sA$

Verification: The signature is verified using a bilinear map as follows:

- $h = hash(message, R)$
- $Q = R + hA$
- $\hat{e}(Q, sP) \stackrel{?}{=} \hat{e}(S, P)$

We are basically checking to see that S is sQ and since we know Q contains Alices public key then S must contain her private key. However it is interesting to note that once we have constructed Q for any signer / message combination the relationship between S and Q is constant, i.e. S is sQ.

The importance of the hash function in Cha & Cheon's identity based scheme

We now examine the importance of the hash function in this signature scheme. Consider a scheme where only the message is hashed and not the message and the random challenge R. In this scenario the following attack would be possible on the signature scheme. Say that S and R represent a genuine signature on a message that hashes to m and we would like to forge a signature on m'.

- $R = rA$
- $S = (r + m)sA$
- $m = m' + a$, for some new message which hashes to m'
- $S = (r + m' + a)sA$

rewrite S

- $S = ((r + a) + m')sA$
- $R' = R + aA = (r + a)A$

Where (R', S) is a valid signature on m'. However the inclusion of R in the hash function of the real Cha & Cheon signature scheme forces the creation of R before S and therefore taking a valid S *(which requires the private key to construct)* and reconstructing a value R for a particular message, as was done above, is not possible.

3.5 Yi's Identity Based Signature

Yi's signature relies on the same Decision Diffie Hellman problem [14]. His signature is essentially the following, though he uses point compression to save on bandwith. The setup and extract algorithms are the same as for the Cha & Cheon scheme above.

Signature: Alice generates a random challenge R. She hashes the message and R to obtain $h \in \mathbb{Z}_r^*$. Alice signs the message by producing the value S below. Again the signature is the pair of points R and S.

- $R = rP$
- $h = hash(message, R)$
- $S = hsA + rsP$

Verification: The signature is verified using a bilinear map as follows:

- $h = hash(message, R)$
- $Q = hA + R$
- $\hat{e}(Q, sP) \stackrel{?}{=} \hat{e}(S, P)$

4 The verification process for the Batch Short Signature Scheme

We have previously shown the batch short signature scheme which verifies as

- $\hat{e}(xM, x^{-1}P) \stackrel{?}{=} \hat{e}(M, P)$

As you can see from the above equation each signature verification requires two pairings - we have to calculate both the right hand side (RHS) and the left hand side (LHS) of the above equation. However we can make this process much simpler. We note the RHS is a pairing that contains a constant point P, this is the basis of our efficiency gain. Now consider that we have the following signatures from Alice, Bob and Carol. Alice's private key is a, Bob's private key is b and Carol's private key is c.

- Alice produces the signature aM and her public key is $a^{-1}P$.
- Bob produces the signature bM' and her public key is $b^{-1}P$.
- Carol produces the signature cM'' and her public key is $c^{-1}P$.

These three signatures can be verified to give

- $\hat{e}(aM, a^{-1}P) \stackrel{?}{=} \hat{e}(M, P)$
- $\hat{e}(bM', b^{-1}P) \stackrel{?}{=} \hat{e}(M', P)$
- $\hat{e}(cM'', c^{-1}P) \stackrel{?}{=} \hat{e}(M'', P)$

But, these verification values multiplied together give

- $\hat{e}(M + M' + M'', P)$

Therefore we can batch verify these signatures as

- $B = M + M' + M''$

- if $B = O$ ***quit*** *(this will happen with* ***negligible*** *probability for an appropriate hash function - perhaps remove one signature and batch verify most of the signatures)*
- $\hat{e}(aM, a^{-1}P) \times \hat{e}(bM', b^{-1}P) \times \hat{e}(cM'', c^{-1}P) \stackrel{?}{=} \hat{e}(B, P)$

In this way we can reduce the number of pairings needed to verify n BLS signatures from $2n$ to $n + 1$.

If we look at pairings in further detail we will see that they are composed of some operation based on Millers algorithm [6], which we will call a "partial pairing" *(this differs for the Weil and Tate pairing)* followed by a final exponentiation by the group order r. Solinas has dubbed this *"Miller lite"* for the Tate pairing when $P \in \mathbb{G}_0$ and the algorithm can be highly optimised [10]. It is interesting to note that

- $Pairing = (partial\ pairing)^r$

therefore

- $Pairing \times Pairing = (partial\ pairing)^r \times (partial\ pairing)^r$

and simplifying gives

- $Pairing \times Pairing = ((partial\ pairing) \times (partial\ pairing))^r$

Therefore, when we are multiplying a series of pairings together we only have to do the final exponentiation once. This allows for an additional saving.

Therefore if we consider that the total computational cost of a pairing is a partial pairing and a final exponentiation, the computational effort for verifying n signatures drops from $((2n \times partial\ pairing) + (2n \times final\ exponentiation))$ to $(((n + 1) \times partial\ pairing) + (2 \times final\ exponentiation))$.

- $(pp(aM, a^{-1}P) \times pp(bM', b^{-1}P) \times pp(cM'', c^{-1}P))^r \stackrel{?}{=} \hat{e}(M + M' + M'', P)$

This equation can be simplified to remove a further final exponentiation.

- $(pp(aM, a^{-1}P) \times pp(bM', b^{-1}P) \times pp(cM'', c^{-1}P)) \times pp(M + M' + M'', -P))^r \stackrel{?}{=} 1$

where $pp(.)$ is the partial pairing operation. The overall cost is now $(((n + 1) \times partial\ pairing) + (1 \times final\ exponentiation))$.

5 The new batch verification for Cha & Cheon's Identity Based Signature Scheme

We have seen that Cha & Cheon's and Yi's identity based signature schemes rely around constructing a point from the persons identity and being given a second point and using Decision Diffie Hellman to verify that the relationship between

these points is that the second is the first multiplied by the master secret. Again consider Alice, Bob and Conor, whose identities map to the points A, B and C respectively.
Alice generates the following signature

- $R = rA$
- $h = hash(message, R)$
- $S = (r + h)sA$

Bob generates the following signature

- $R' = r'B$
- $h' = hash(message', R')$
- $S' = (r' + h')sB$

Carol generates the following signature

- $R'' = r''C$
- $h'' = hash(message'', R'')$
- $S'' = (r'' + h'')sC$

Now consider the individual verification of these signatures.
For Alice's signature we have

- $h = hash(message, R)$
- $Q = R + hA$
- $\hat{e}(Q, sP) \stackrel{?}{=} \hat{e}(S, P)$

For Bob's signature we have

- $h' = hash(message', R')$
- $Q' = R' + h'B$
- $\hat{e}(Q', sP) \stackrel{?}{=} \hat{e}(S', P)$

For Carol's signature we have

- $h'' = hash(message'', R'')$
- $Q'' = R'' + h''C$
- $\hat{e}(Q'', sP) \stackrel{?}{=} \hat{e}(S'', P)$

But if we multiply together the LHS's of these verification equations we will get the following:

- $\hat{e}(Q + Q' + Q'', sP) \stackrel{?}{=} \hat{e}(S, P) \times \hat{e}(S', P) \times \hat{e}(S'', P)$

Simplifying this equation we get

- $\hat{e}(Q + Q' + Q'', sP) \stackrel{?}{=} \hat{e}(S + S' + S'', P)$

And again applying the optimisation mentioned in the previous section.

$$- \ (pp(Q + Q' + Q'', sP) \times pp(S + S' + S'', -P))^r \stackrel{?}{=} 1$$

Therefore we have reduced the computational effort required to verify n signatures from $2n$ pairings to $(2 \times \mathit{partial\ pairing}) + (1 \times \mathit{final\ exponentiation})$.

This same method works for speeding up the verification of Yi's identity based signature scheme. However, it is also interesting to note that we can mix both of these signature schemes in one batch verification process, provided all private keys have been issued by the same KGC.

6 Conclusion

We have looked again at three signature schemes based on bilinear maps and we have seen how we can use the properties of bilinear maps to speed up the batch verification of these signature schemes. We have shown that identity based signatures benefit most from batch verification as there is a constant relationship between the public and private keys in identity based cryptosystems which use pairings.

Acknowledgements: The author would like to thank Michael Scott and Paulo Barreto for their help in writing this paper.

References

1. M. Bellare, J. Garay, and T. Rabin. Fast batch verification for modular exponentiation and digital signatures. In *Proceeding of Eurocrypt 98, LNCS Vol. 1403, K. Nyberg ed., Springer-Verlag, 1998*, 1998. http://www.cs.ucsd.edu/users/mihir/papers/batch.pdf.
2. D. Boneh, H. Shacham, and B. Lynn. Short signatures from the weil pairing. In proceedings of Asiacrypt '01, LNCS Vol. 2139, Springer-Verlag, pp. 514-532, 2001.
3. Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing, 2001. http://citeseer.nj.nec.com/boneh01identitybased.html.
4. J. Cha and J. Cheon. An identity-based signature from gap diffie-hellman groups. Cryptology ePrint Archive, Report 2002/018.
5. A. Fiat. Batch rsa. Journal of Cryptology, Vol. 10, No. 2, 1997. http://www.math.tau.ac.il/~fiat/batrsa.ps.
6. Victor Miller. Short programs for functions on curves. Unpublished Manuscript, 1986. http://citeseer.nj.nec.com/cache/papers/cs/26957/http:zSzzSzcrypto.stan%ford.eduzSzmillerzSzmiller.pdf/miller86short.pdf.
7. D. M'Raïhi and D. Naccache. Batch exponentiation - a fast dlp based signature generation strategy. In *Proceeding of 3rd ACM Conference on Computer and Communications Security, ACM, 1996*, 1996. http://www.gemplus.com/smart/r_d/publications/pdf/MN96batc.pdf.
8. D. Naccache, D. M'Raïhi, S. Vaudenay, and D. Raphaell. Can dsa be improved? complexity trade-offs with the digital signature standard. In *Proceeding of Eurocrypt 94, LNCS Vol. 950, Springer-Verlag, A. De Santis ed., 1994*, 1994. http://www.gemplus.com/smart/r_d/publications/pdf/NMVR94sa.pdf.

9. Michael Scott. The tate pairing. 2003. http://www.computing.dcu.ie/~mike/tate.html.
10. Jerome A. Solinas. Id-based digital signature algorithms. Slide Show presented at 7th Workshop on Elliptic Curve Cryptography.
11. The IDC staff. Email mailboxes to increase to 1.2 billion worldwide by 2005. CNN.com. http://www.cnn.com/2001/TECH/internet/09/19/email.usage.idg/.
12. Yacov Yacobi. A note on the bilinear diffie-hellman assumption. Cryptology ePrint Archive, Report 2002/113, 2002. http://eprint.iacr.org/2002/113.
13. S. Yen and C. Laih. Improved digital siganture suitible for bathc verification. IEEE Transactions on Computers, Vol. 44, No. 7, 1995, 1995. http://crypto.ee.ncku.edu.tw/pdf/iel_cgi-15.pdf.
14. Xun Yi. An identity based signature scheme from the weil pairing. IEEE Communcations Letters, volume 7, no. 2, February 2003, pp. 76-78.
15. Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo. An efficient signature scheme from bilinear pairings and its applications. In *Proceeding of PKC 2004, Singapore. LNCS, Springer-Verlag.* http://www.uow.edu.au/~fangguo/PKC04.pdf.