6.897: Selected Topics in Cryptography May 13, 2004

Lecture 26

Lecturer: Ron Rivest *Scribe: Dah-Yoh Lim*

1 Recap of Pairing-Based Cryptography

Setting (as usual): $(G_1 = \langle P \rangle, +)$,$(G_2, \cdot)$: two groups of the same prime order q. Assume DLP is hard in both groups. Note that in the literature sometimes both G_1 and G_2 are multiplicative groups.

Definition 1. *A mapping $e : G_1 \times G_1 \to G_2$ is called a bilinear map iff:*

1. *Bilinearity: $\forall P, Q \in G_1, \forall a, b \in \mathbb{Z}_q^*, e(aP, bQ) = e(P, Q)^{ab}$.*
2. *Nondegeneracy: $G_1 = \langle P \rangle \implies \langle e(P, P) \rangle = G_2$; equivalently, $P \neq 0 \implies e(P, P) \neq 1$.*
3. *Computable in polynomial time.*

Theorem 2. *DDH is easy in G_1.*

Proof. Given a quadruple (P, aP, bP, cP), test whether $e(P, cP) = e(aP, bP)$. Equality holds iff $ab = c$.

□

Last time we saw the result of Joux [Jou00] on one-round 3-party key agreement, where each party i sends a_iP to the other parties. They then compute $e(P, P)^{a_1,a_2,a_3}$. This is secure against passive adversaries if the Bilinear Diffie-Hellman Assumption holds, which says that it is hard to compute $e(P, P)^{a_1,a_2,a_3}$ from (P, a_1P, a_2P, a_3P). The paper also gives a way to extend this to multiparty key agreement, basically by building a trinary tree.

Also, recall that we saw the identity-based encryption scheme of Boneh and Franklin [BoFr01].

2 The BLS Signature scheme

In [BLS01], Boneh, Lynn and Shacham gave a simple, deterministic signature scheme where the signatures are very short. Security is proven under the random-oracle model.

The signer's secret key is $x \in_R \mathbb{Z}_q^*$; the public key is $y = g^x$, an element in G_1. Note that $G_1 = \langle g \rangle$ is multiplicative here. Let $H : \{0, 1\}^* \to G_1$ be a hash function.

$\mathsf{Sign}(m)$: the signature σ on message m is $H(m)^x$ (in G_1).

$\mathsf{Verify}(\sigma, m)$: accept iff $(g, y, H(m), \sigma)$ is a Diffie-Hellman quadruple which will be the case iff $e(g, \sigma) = e(y, H(m))$.

Security: Unexistentially forgeable under adaptive chosen message attack in the random oracle model, assuming that the CDH is hard on certain elliptic curves over a finite field of characteristic 3.

Efficiency: signing is fast, only one hashing operation and one modular exponentiation. Verification requires two pairing computations, which according to [BLS01] takes 3 seconds on a PIII 1 GHz computer, where the computations are over an elliptic curve on $\mathbb{F}_{3^{97}}$. The signature is just an element in G_1, which is 154 bits if we use an elliptic curve on $\mathbb{F}_{3^{97}}$. This is just half the size of the signature in DSA (320 bits) with comparable security. This makes the BLS scheme the current scheme with the shortest signature.

Barreto, Kim and Scott [BKS03] gave a faster implementation and compared the resulting times with that using RSA. Under RSA with $|n|$=1024 bits, $|d| = 1007$ bits, signing took 8 ms and verifying took 400 μsec. Under the BLS using elliptic curves over $\mathbb{F}_{3^{97}}$, signing took 3.5 ms and verifying took 53 ms. Therefore the BLS scheme is quite practical.

Question: is it possible to use this scheme to do blind signatures? Probably yes, by having a setup phase in which the signer is asked to sign messages $m_1, ..., m_k$, giving the pairs $H(m_i), H(m_i)^x$. To get a blind signature on the message m, first compute the blinding factor r (and also r^x), as the product of a random subset of the $H(m_i)$s ($H(m_i)^x$s, respectively). The idea is to get some random r such that you know r^x, so when the signer signs $rH(m)$ you can get something like $r^x H(m)^x$, where m is the message for which you want to get a blind signature on; then you divide out by r^x and you get the signature on m. However, there is a problem with this, namely that with such a random r and a message m, what message do you ask the signer to sign? (The signer doing blind signatures now merely needs to raise its input to the x-th power.)

Various extensions to the basic BLS scheme has been studied, and we will discuss some of them in the following sections.

3 Multisignature [Bol03]

Consider the setting in which several signers all wish to sign the same message m.

Signer i's secret key is $x_i \in_R \mathbb{Z}_q^*$ and public key $y_i = g^{x_i}$.

Sign(m): the signature σ on message m is $\sigma = \Pi_i \sigma_i$, where σ_i is the BLS signature of signer i on message m. In other words, $\sigma = H(m)^{\sum_i x_i}$. More precisely, the signature should be $(\sigma, y = \Pi_i y_i,$ list of signers).

Verify($m, signature$): as in the BLS, accept iff $e(g, \sigma) = e(y, H(m))$.

Security is exactly as in the BLS. Note that this is really just an n-out-of-n threshold version of BLS.

4 Threshold signatures [Bol03]

Now consider the more general t-out-of-n threshold signatures, i.e. any t signers in the group of n signers can sign a message m.

Here, the master secret $= x = \sum_i x_i L_i$, where the L_i's are the Lagrange coefficients, which depends on which t values of i you have. The master public key is $y = g^x$, and the individual public

keys $y_i = g^{x_i}$.

Sign(m): first, each of the t signers creates his share of the signature: $\sigma_i = H(m)^{x_i}$. Note that everyone can check the validity of σ_i by checking $e(g, \sigma_i) = e(y_i, H(m))$ (as in the plain BLS scheme). The signature $\sigma = \Pi_i \sigma_i^{L_i}$ for t values of i.

Verify(m, σ): accept iff $e(g, \sigma) = e(y, H(m))$.

5 Aggregate Signatures [BGL+03]

This further generalizes threshold signatures. In this case we have different signers that want to sign different messages, but we only want to produce one signature. This is useful for instance in cases where it is natural to validate different signatures on different messages by different signers in one shot.

Signer i has secret key x_i and public key $y_i = g^{x_i}$, and wishes to sign message m_i, where for technical reasons (and without loss of generality) assume that all the m_i's are distinct.

Sign$(m_1, ..., m_n)$: first, each signer computes its signature: $\sigma_i = H(m_i)^{x_i}$, $1 \leq i \leq n$. The aggregate signature is $\sigma = \Pi_i \sigma_i$.

Verify(m, σ): accept iff $e(g, \sigma) = \Pi_{i=1}^{n}\ e(y_i, H(m_i))$.

This scheme is secure against existential forgery with chosen message attacks if the computational Co-DH problem is hard: given g, g^a (in G_1), and h (in G_2), it is hard to compute h^a (in G_2).

An application of aggregate signatures is in certificate chains where say party A signs the message "B is one of my children along the chain, and he has public key PK_B", i.e. A gives a certificate on B's public key. Similarly, B might then give a certificate on C's public key, and so on. So for some one down the line, say D, to prove that his public key is PK_D, he just has to give A's certificate on B,B's certificate on C and C's certificate on D together with the aggregate signature, which the verifier can verify in one shot that all the certificates are valid.

6 Bilinear Ring Signatures

Here, any one party can create a signature from his own secret key and the public key of the others, without having to cooperate with them, i.e. there is no setup phase required. They might not even know that the signer s created a ring signature that includes them. The verifier cannot tell who in the group actually signed the document (this is called signer anonymity).

Let $G_1 = \langle g \rangle$ and $H : \{0,1\}^* \to G_1$. Signer i has secret key x_i and public key $y_i = g^{x_i}$.

Sign$(y_1, ..., y_{s-1}, y_{s+1}, ..., y_n, m, x_s)$: $\forall i \in [1, ..., n], i \neq s$ chose $r_i \in_R \mathbb{Z}_q^*$ and compute $\sigma_i = g^{r_i}$. Compute $\sigma_s = (\frac{H(m)}{\Pi_{i \neq s} y_i^{r_i}})^{1/x_s}$. The signature is $(\sigma_1, ..., \sigma_n)$.

Verify : accept iff $e(g, H(m)) = \Pi_{i=1}^{n}\ e(y_i, \sigma_i)$. This holds true for good signatures because the right-hand side can be simplified into:

$$\Pi_{i\neq s}\ e(g^{x_i}, g^{r_i})e(g^{x_s}, (\frac{H(m)}{\Pi_{i\neq s} y_i^{r_i}})^{1/x_s}) = \Pi_{i\neq s}\ e(g, g^{x_i r_i})e(g, (\frac{H(m)}{\Pi_{i\neq s} g^{x_i r_i}})) = e(g, H(m)).$$

Signer anonymity is unconditional. Security against forgery is proven in the random oracle model, where the adversary has access to H and a ring-signing oracle, under the CDH assumption.

References

[Jou00] A. Joux. A One Round Protocol for Tripartite Diffie-Hellman. Proceedings of ANTS 4, LNCS 1838, pp. 385-394, 2000.

[BoFr01] D. Boneh, M. Franklin. Identity Based Encryption from the Weil Pairing. SIAM J. of Computing, Vol. 32, No. 3, pp. 586-615, 2003, Extended abstract in Crypto 2001.

[BLS01] D. Boneh, B. Lynn, H. Shacham. Short Signatures from the Weil Pairing. in Proceedings of Asiacrypt 2001.

[BKS03] P. S. L. M. Berreto, H. Y. Kim and M. Scott. Efficient Algorithms for Pairing-Based Cryptosystems. Advances in Cryptology - Crypto 2002, LNCS 2442, pp. 354-368.

[Bol03] A. Boldyreva. Efficient Threshold Signature, Multisignature and Blind Signature Schemes Based on the Gap-Diffie-Hellman-Group Signature Scheme. PKC 2003, LNCS 2139, pp. 31-46, Springer-Verlag, 2003.

[BGL+03] D. Boneh, C. Gentry, B. Lynn and H. Shacham. Aggregate and Verifiably Encrypted Signature from Bilinear Maps. Eurocrypt 2003, LNCS 2248, pp. 514-532, Springer-Verlag, 2003.