A SHORT NOTE ON SECRET SHARING USING ELLIPTIC CURVES

Volker Müller
University of Luxembourg, Faculty of Sciences, Technology and Communication
6, rue Richard Coudenhove-Kalergi, L-1359, Luxembourg

Keywords: Elliptic curve, threshold scheme, verifiable secret sharing, bilinear map.

Abstract: In this short note, we describe a variant of Shamir's (n,t)-threshold scheme based on elliptic curves. Moreover, we show how pairings of elliptic curves can be used to also provide verifiability for the new elliptic curve based threshold scheme.

1 INTRODUCTION

Sharing a secret between a group of participants is a well-known and long solved problem in cryptography. A (n,t)-threshold scheme is a method by which a trusted third party computes n secret shares from a secret and distributes these shares secretly to the n participants. If t or more participants pool their shares, then the secret can be determined, otherwise no substantial information about the secret is given (Menezes et al., 1997). Shamir first described a (n,t)-threshold scheme based on polynomial interpolation over finite fields (Shamir, 1979). In this short note, we describe how the ideas of Shamir's threshold scheme can be slightly modified to obtain a (n,t)-threshold scheme based on elliptic curves. An additional property of this new scheme is the fact that any already existing elliptic curve related cryptographic information can be reused and existing security devices like smartcards can easily be adapted to the new threshold scheme.

We assume that the reader is already familiar with elliptic curves and their usage in public key cryptography; descriptions of ECC in theory and practice can be found in, e.g., (Hankerson et al., 2004), (Koblitz, 1987), (Certicom, 2000), and many other publications. In the following, we assume that K denotes a finite prime field with q elements, and E is a cryptographically secure elliptic curve defined over K. The group of points on E defined over K is denoted as $E(K)$.

2 SECRET SHARING USING ELLIPTIC CURVES

Shamir's scheme for secret sharing (Shamir, 1979) uses polynomial arithmetic and interpolation. The scheme encodes a secret as the constant term of an otherwise randomly chosen polynomial $f(x)$ of degree $t-1$ defined over a fixed finite field K. A share of the secret is then a pair $(x_i, f(x_i)) \in K^2$. The first component x_i of this share can even be made public and directly depend on the identity of the corresponding participant, but the second component $f(x_i)$ must be absolutely kept secret. Any t different such pairs are sufficient to reconstruct the secret using polynomial interpolation; on the other hand, the knowledge of less than t pairs does not yield the polynomial f, and therefore does not open the shared secret.

There exist several algorithms for polynomial interpolation over a field (see, e.g., (Stoer and Burlirsch, 1991)). Using the polynomial $\omega(x) = \prod_{j=1}^{t}(x-x_j)$, the Lagrange interpolating polynomial $f(x)$ for t pairs $(x_i, f(x_i)), 1 \le i \le t$, is given as

$$f(x) = \sum_{i=1}^{t} \frac{\omega(x)}{(x-x_i)\cdot\omega'(x_i)} \cdot f(x_i)\,. \qquad (1)$$

Interestingly for elliptic curves, formula (1) is linear in $f(x_i)$, and therefore easy to apply also in the group of points on an elliptic curve. We assume that from an ECC setup we already know a cryptographically strong elliptic curve E defined over a finite prime field K of q elements and a base point $P \in E(K)$ with order larger than q. For simplicity, we assume that $E(K)$ is cyclic, and P is a generator of the group. Addition-

Müller V. (2008).
A SHORT NOTE ON SECRET SHARING USING ELLIPTIC CURVES.
In *Proceedings of the International Conference on Security and Cryptography*, pages 359-362
DOI: 10.5220/0001918303590362

ally, for every participant i in the threshold scheme there exists a public key point $Q_i = d_i \cdot P$, where the integer $0 < d_i < \text{ord}(P)$ defines the secret key of that participant.

The general idea of the elliptic curve (n,t)-threshold scheme is based on the fact that with (1) we can determine $f(\lambda) \cdot P$ for P and any integer $0 \leq \lambda < q$ if we know t different points $f(x_i) \cdot P$ for modulo q pairwise different integers $x_i \not\equiv 0 \bmod \text{ord}(P)$. Therefore, the trusted third party can set up the system by choosing a random polynomial $f(x) \in K[x]$ of degree $t-1$, and secretly distributing the shares $(x_i, f(x_i) \cdot P), 1 \leq i \leq n$, to the n participants. The n integers $0 < x_i < q$ must be pairwise different, but as in Shamir's system they can be made public or directly computable from the identity of the participants. Secure distribution of the secret part $f(x_i) \cdot P$ of the shares to the participants can be done by encrypting it with the ECC public key of the corresponding participant. Then this ciphertext is either communicated to that participant over an insecure channel, or it can be published, since only the owner of the correct ECC secret key can open that partial share. When at least t participants pool their shares, then they can determine the point $f(0) \cdot P$ using (1). In contrast to Shamir's system, we do not encode the global secret m as one of the coordinates of a point, but we use $f(0) \cdot P$ as a secret key for some fixed secret key cryptosystem to encrypt m. More precisely, we proceed as in the Elliptic Curve Integrated Encryption Scheme (e.g., (Certicom, 2000)) and apply a secret key cryptosystem ENC, a key derivation function KDF and a message authentication code MAC to first find $k_E || k_M = KDF(x(f(0) \cdot P))$ and then publish the encrypted secret as $c||d$ where $c = ENC(k_E, m)$ and $d = MAC(k_M, c)$. It is obvious that anybody who can determine the secret point $f(0) \cdot P$ can also easily open the encrypted global secret by first computing k_E and k_M and then applying the secret key decryption procedure.

Theorem 1. *Knowledge of t or more shares opens the global secret m. On the other hand, knowledge of less than t shares only yields at least $q/2$ many possibilities for the input of the KDF if the order of P is greater than q.*

Proof: The proof is essentially equal to the proof of Shamir's system. As described above, the point $f(0) \cdot P$ can be determined easily with polynomial interpolation for t or more known shares. On the other hand, there are q possible constant terms for polynomials of degree $t-1$ given at most $t-1$ pairs $(x_i, f(x_i))$. If the order of P is greater than q, then this leads to q possibilities for the point $f(0) \cdot P$. Since we are using only the x-coordinate of that point to encrypt the global secret, there remain at least $q/2$ many possible inputs to the KDF. □

It should be noted that KDF and ENC should be chosen with appropriate parameters (especially providing a sufficiently large key space for ENC) since otherwise the total system will be insecure. After the setup of the threshold scheme, the following protocol can be started by a dedicated participant (with index 1) to open the shared secret with the help of $t-1$ other participants:

- Participant 1 chooses a random point $H \in E(K)$, decrypts his encrypted share $f(x_1) \cdot P$ using his secret ECC key and determines with his share the result $H - \frac{\omega(0)}{x_1 \cdot \omega'(x_1)} \cdot (f(x_1) \cdot P)$. Then he sends this information to the next participant. Note that if all values x_i are publicly known, then $\omega(0)$ and $\omega'(x_i)$ can be precomputed.
- The second participant decrypts his secret share $f(x_2) \cdot P$ with his secret ECC key, subtracts the point $\frac{\omega(0)}{x_2 \cdot \omega'(x_2)} \cdot (f(x_2) \cdot P)$ from his input point and sends the result to the next participant. All other participants do the same with their shares, respectively. The last participants forwards the result to participant 1 that started the whole protocol.
- Participant 1 subtracts the randomly chosen initial point H from his input point and obtains the secret point $f(0) \cdot P$. He can then open the global secret.

The proof that this scheme really determines $f(0) \cdot P$ directly follows from (1). Note that this EC threshold scheme is neither ideal nor perfect, but nevertheless it is practical since it does not require knowledge of any additional secret key.

3 VERIFIABLE SECRET SHARING VARIANTS

In the last 10 years, bilinear maps for elliptic curves (also denoted pairings) have been applied to various cryptographic applications (CL, 2008). We can also use such maps for the EC (n,t)-threshold scheme to provide additional properties. Assume that for a given cryptographically strong elliptic curve E there exists a some small positive integer s and a bilinear map e

$$e: E(K) \times E(K) \longrightarrow K^s; \quad e(a \cdot P, b \cdot Q) = e(P,Q)^{ab}$$

with the additional property that for points $P \neq \mathcal{O}$ we have $e(P,P) \neq 1$. Such maps are for example given by the Weil pairing or the Tate pairing (Galbraith et al., 2002). The importance of these maps for cryptographic applications is the fact that they "link" the discrete logarithm in the elliptic curve point group to

a discrete logarithm in the finite field K^s. Therefore, the security of the ECC system enforces certain conditions on E and K such that the discrete logarithm problem in K^s is also difficult to solve. In the following, we will make use of such maps to add verification procedures to the EC threshold scheme described above (note that similar techniques were also used in (Baek and Zheng, 2004) and (Liu et al., 2007)).

3.1 Verifiable Secret Sharing à La Feldman

We describe an EC variant for the verifiable secret sharing scheme of Feldman (Feldman, 1987), where additional information (so called commitments) is provided such that the participants can verify the correctness of their shares.

The commitments in the EC variant of Feldman's scheme are given as the field elements $e(P,P)^{a_i} \in K^s, 0 \leq i \leq t-1$, where the a_i are the coefficients of the secret polynomial $f(x)$ used for the construction of the shares. These commitments are published by the trusted third party after the system setup. Using these commitments, every participant can determine for any $\lambda \in K$ the value

$$e(P,P)^{f(\lambda)} = \prod_{i=0}^{t-1} \left(e(P,P)^{a_i} \right)^{\lambda^i}. \qquad (2)$$

Therefore, the j-th participant can determine $e(P,P)^{f(x_j)}$ in two ways: either with (2), or by using his private share $f(x_j) \cdot P$ and a pairing computation. If both values should be different, then either his private share was wrong, or the trusted third party cheated with the publication of the values $e(P,P)^{a_i}$.

Lemma 1. *If the two results are equal and the trusted third party did not cheat, then the private share of the j-th participant really equals $f(x_j) \cdot P$.*

Proof: Assume that the share of the j-th participant is incorrect, i.e. he receives a point $\lambda \cdot P$ for some integer $\lambda \neq f(x_j)$, but nevertheless the test above succeeds. Then $e(P,P)^{f(x_j)} = e(P,P)^{\lambda}$, or equivalently, $e(P,P)^{f(x_j)-\lambda} = 1$. So $f(x_j) \equiv \lambda \bmod \mathrm{ord}(P)$, and $\lambda \cdot P = f(x_j) \cdot P$, a contradiction. □

With (2), it is obvious that everybody can determine $e(P,P)^{f(\lambda)}$ for every integer $0 \leq \lambda < q$. The pairing inversion problem is defined as the problem to compute for given value $e(P,H)$ a suitable point H. If the pairing inversion problem were easy, then it would be also easy to determine individual shares for non-legitimate users – just determine the field element $e(P,P)^{(f(x_j)}$ with (2) and solve the corresponding pairing inversion problem. This would break the complete EC threshold scheme. However, pairing inversion seems in general to be hard (Galbraith et al., 2008).

Therefore, practical parameters for the EC threshold scheme should be chosen such that no "simple" algorithm for the pairing inversion problem is known for the used elliptic curve.

3.2 Distributing the Global Secret to All other Participants

The protocol presented in the last section was started by some dedicated participant. That participant needed the help of at least $t-1$ other participants to determine the point $f(0) \cdot P$ and so open the global secret m. A disadvantage of this protocol is the fact that only one out of the t involved participants finally knows m. Using the commitments defined above, the dedicated participant can announce the point $f(0) \cdot P$ to all other participants, of course encrypted with the individual secret EC keys of the other participants. Any participant can then use the commitments to determine the value $e(P,P)^{f(0)}$ using (2) and a pairing computation with the received point, such that he can verify the correctness of the information he received from the dedicated first participant. Of course, knowledge of the point $f(0) \cdot P$ is also sufficient to determine the global secret. Note that directly sending an encrypted version of m to all other participants does not given them the possibility to verify the correctness of m.

3.3 Verifying Intermediate Results

We can extend the verifiability described in the last section such that even the validity of all intermediate results can be verified. In this variant, a cheating participant (i.e. a participant that does not apply his own private share) can be determined. We extend the protocol given in Section 2 such that every participant publishes an own commitment of his contribution. Remember that the j-participant in the protocol forwards the point $R_j = H - \sum_{i=1}^{j} \frac{\omega(0)}{x_i \cdot \omega'(x_i)} \cdot (f(x_i) \cdot P)$ to the next participant. The commitments of the participants are then given as follows: the initial participant publishes his commitment $e(P,H)$ and $e(P,R_1)$, whereas all other participant add their own commitments as $e(P,R_j)$.

Using these participant commitments, it is easy to check the validity of each intermediate result:

$$e(P,R_j) = e(P,H) \cdot \prod_{i=1}^{j} \left(e(P,P)^{f(x_i)} \right)^{-\omega(0)/(x_i \cdot \omega'(x_i))}.$$

Since the dedicated participant that starts the protocol is interested in obtaining the global secret, he should have no interest in cheating, and we assume that he is honest.

Theorem 2. *If the first participant is honest, then the identity of any cheating participant can be determined from the participant commitments.*

Proof: During the protocol, every participant compares the pairing value determined with the input point he received from the previous participant with that participant's commitment. If both pairing values do not match, then obviously that participant was cheating, and the protocol exits with error. Note also that

$$e(P,R_j) = e(P,R_{j-1}) \cdot \left(e(P,P)^{f(x_j)}\right)^{-\omega(0)/(x_j \cdot \omega'(x_j))}, \quad (3)$$

such that the correctness of the j-th commitment depends directly on the correctness of the previous commitment (note that the second term in the product can be computed using the trusted third party's commitments). Therefore, the commitment of the first participant can be used to successively verify the correctness of all other participants' commitments such that a cheating participant j must publish his correct commitment $e(P,R_j)$. Assume that he cheats by forwarding a wrong intermediate point $R'_j \neq R_j$ to the next participant. Since $e(P,R'_j) = e(P,R_j)$ implies $e(P,R'_j - R_j) = 1$ or $R'_j = R_j$ (note that the group of points is cyclic), this will be detected by participant $j+1$ when he compares the two possibilities for $e(P,R_j)$ determined with (3) and with a pairing computation based on his two input points P and R'_j. □

4 CONCLUSIONS

In this short note we have presented a simple generalization of Shamir's (n,t)-threshold scheme based on elliptic curves and three variants of it that use bilinear maps. This EC threshold scheme needs no additional secret keys, since it reuses existing public and secret ECC keys. It can therefore be directly used with existing EC security devices.

REFERENCES

Baek, J. and Zheng, Y. (2004). Identity-based threshold decryption. In *PKC 2004, LNCS 2947*, pages 262–276.

Certicom (2000). Standards for efficient cryptography, sec 1: Elliptic curve cryptography, available at http://www.secg.org/download/aid-385/sec1_final.pdf.

CL (2008). The pairing-based crypto lounge, website at http://paginas.terra.com.br/informatica/paulobarreto/pblounge.html.

Feldman, P. (1987). A practical scheme for non-interactive verifiable secret sharing. In *IEEE Symposium on Foundations of Computer Science*, pages 427–437.

Galbraith, S., Harrison, K., and Soldera, D. (2002). Implementing the tate pairing. In *Algorithmic Number Theory Symposium – ANTS-V, Lecture Notes on Computer Science*, volume 2369, pages 324–337. Springer.

Galbraith, S., Hess, F., and Vercauteren, F. (2008). Aspects of pairing inversion. Technical report, Katholieke Universiteit Leuven, available at http://homes.esat.kuleuven.be/∼fvercaut/.

Hankerson, D., Menezes, A., and Vanstone, S. (2004). *Guide to Elliptic Curve Cryptography*. Springer.

Koblitz, N. (1987). Elliptic curve cryptosystems. In *Mathematics of Computation*, volume 48, pages 203–209.

Liu, S., Chen, K., and Qiu, W. (2007). Identity-based threshold decryption revisited. In *ISPEC 2007, LNCS 4464*, pages 329–343.

Menezes, A., Oorschot, P., and Vanstone, S. (1997). *Handbook of Applied Cryptography*. CRC Press.

Shamir, A. (1979). How to share a secret. In *Communications of the ACM*, volume 22, pages 612–613.

Stoer, J. and Burlirsch, R. (1991). *Introduction to Numerical Analysis*. Springer.