Requirements for Standard Elliptic Curves

Position Paper of the ECC Brainpool[1]

Dr. Manfred Lochter, Bundesamt für Sicherheit in der Informationstechnik, manfred.lochter<at>bsi.bund.de

Dr. Johannes Merkle, secunet Security Networks, johannes.merkle<at>secunet.com

Dr. Jörn-Marc Schmidt, secunet Security Networks, joern-marc.schmidt<at>secunet.com

Dr. Torsten Schütze, Rohde & Schwarz SIT, Torsten.Schuetze<at>rohde-schwarz.com

September 30, 2014

Abstract

Currently, the Internet Research Task Force (IRTF) discusses requirements for new elliptic curves to be standardized in TLS and other internet protocols. This position paper discusses the view of the members of the ECC Brainpool on these requirements, in particular with respect to hardware implementations.

Recently, the TLS working group of the Internet Engineering Task Force (IETF) has formally requested the IRTF Crypto Forum Research Group (CFRG) to recommend one or more sets of elliptic curve parameters for standardization in TLS (for key agreement and authentication) and potentially other internet protocols [1]. Following this request, a discussion has started within CFRG on the requirements for the new curves. The ECC Brainpool working group[2] took this opportunity to discuss their view on the requirements and desired properties of future standard curves.

On the CFRG mailing list, users of elliptic curve cryptography (ECC) in software expressed their demand for high performance implementations. That led to a discussion focusing on special curves and prime structures, which allow specific optimizations and, hence, fast software implementations. Actually, the discussion should not be about software versus hardware, but software in a secure environment, e.g., on a secured server, where only the global timing as side-channel is relevant versus software on constrained devices or hardware in a hostile environment, where the full set of local and global side-channels applies. The ECC Brainpool working group is convinced that hardware or high-assurance software requirements should rank equal for the following reasons:

1. With the advent of the Internet of Things, even smallest devices will need to support TLS. Those devices may require hardware implementations of ECC operations. Further, we will have demands for more and more high-assurance ECC on constrained devices such as in the smart metering scenario (products used in these scenarios generally have longer life cycles than software implementations and a higher / different attack potential has to be taken into account).
2. In view of actual security incidents like Heartbleed, it is advisable to transfer critical cryptographic operations to specialized hardware modules to protect private keys from exposure by implementation flaws.
3. Finally, the choice and recommendation of the CFRG of parameters for TLS will have a signalling effect for other protocols and applications, including use cases for hardware modules and smart cards.

Against this background, the group discussed several different aspects. Common positions on many aspects were achieved, but that there were also slightly different views on certain aspects.

---

1 The paper results from a meeting of the ECC Brainpool working group, in Bonn, Germany, on September 3rd, 2014.

2 http://www.ecc-brainpool.org

Curve Representation and Exchange Format

Current standards like those of ISO, ANSI, IETF, BSI, and NIST all describe elliptic curves in short Weierstraß form. In addition, data structures for exchanging points of a curve in protocols are also specified as either affine coordinates or compressed affine coordinates for short Weierstraß curves.

In the current discussion, different proposals like Montgomery curves and (twisted) Edwards curves are considered since they allow simple and efficient arithmetic. Points on curves in either of these forms can be efficiently transformed to affine coordinates in Weierstraß form and vice versa. This allows using Montgomery and (twisted) Edwards curves in implementations while still using the exchange formats defined in current standards. In our view, this approach reduces the implementation costs as compared to adapting the exchanging format and is, thus, strongly preferable in case that IETF decides to standardize Montgomery or (twisted) Edwards curves.

Using Special Primes

The use of primes with sparse binary representation, in particular, Pseudo-Mersenne primes, as field size enables optimizations of the field arithmetic. Since software allows short design cycles and fast adoption to new developments, such optimizations can, at least in the non-embedded world, be integrated rather fast. This is different for high-security software on embedded devices and hardware. While hardware implementations can also be optimized for the used prime and try to gain some additional speed in a singular case, this potential gain in performance comes at cost of flexibility: Changing the prime requires a redesign of the hardware. Modifications in the field are not possible. Further, that approach demands for different implementations of the multiplier for ECC and RSA in devices that have to support both algorithms. In case the hardware implementation should be flexible, i.e., support arbitrary primes, a special shape of the prime does not improve performance in hardware. Moreover, it has negative influence on the implementation security aspect, as it hinders efficient randomization countermeasures: A prime field size with sparse binary representation requires larger blinding factors for its randomization as well as for the randomization of the secret scalar, because, by the Hasse's theorem, it also yields a sparse representation of the curve order.

Furthermore, the implementation of an optimized arithmetic / modular multiplier for (additional) special primes in hardware is a major investment for any manufacturer and, thus, comes with high financial risks.

There was a strong consensus in the working group that verifiably pseudo-random primes should be preferred (for hardware and high-assurance[3] software), or that at least such a set of curves should be mandatory to implement.

Implementation Security

The selection of the curve parameters influences the effort for secure implementations. In this context, the required protection depends on the application scenario. In cases where the implementation runs in a secure environment and only remote attacks are possible, a time-constant implementation may be sufficient. If the implementation has to withstand adversaries with physical access, i.e., to be resistant against other side-channels like power consumption and electromagnetic emanations or even against active fault injection, a combination of more advanced protection techniques like randomization is essential [2]. While realizations of constant time implementations are possible for Weierstraß curves as well, a common argument for the use of Montgomery/Edwards curves is that achieving constant time computations is easier. Nevertheless, it should be noted that this is not sufficient for protection against adversaries considered in the second case.

Twist Security

There were different positions on twist security. For an elliptic curve, each element of the underlying

[3] We consider certification according to a commonly accepted certification scheme, e.g., the Common Criteria, as a requirement for high-assurance devices.

field can either be mapped to a point on the curve or on its twist. Hence, in case the twist of a curve has a smooth order, an adversary may try to make the implementation compute a scalar multiplication on the twist instead of the original curve. In order to prevent such an attack, a curve with a secure (high order) twist can be chosen. The use of such curves might improve the security of careful implementations.

Nonetheless, even with twist security receivers must check group membership of publicly communicated EC points. Further, curves with a strong twist constitute a distinguished subset of all possible curves. Even though there is currently no indication for a security issue, some members fear that belonging to a small special class might foster future attacks.

Cofactor

A cofactor greater one requires an additional check or an additional multiplication to prevent small-subgroup attacks. Since this is not only an additional overhead but also a source for implementation weaknesses if the check is omitted or just forgotten[4], a cofactor of one is desirable.

Rigidity

Recent revelations on manipulations of cryptographic standards have raised the demand for a transparent and traceable process on how to select curve parameters. One approach to achieve this is using a pseudo-random generation process which is seeded by natural constants. For example, the *Brainpool Curves* [3] have been generated this way. The second possibility is to define a set of desired properties and to choose out of the remaining options the smallest set, i.e., those with the best performance. This is how, e.g., the *Curve25519* [4] and the *NUMS curves* [5] have been constructed.

Both processes allow very limited flexibility. Nevertheless, the choice of input parameters as well as the choice of desired properties influences the result. Hence, perfect rigidity, i.e., defining a process that is accepted as completely transparent and traceable by everyone, seems to be impossible. Following some of the recent discussions and contributions on selecting elliptic curves, see e.g. [6], the members of the ECC Brainpool working group see the great risk that trustworthy curves, which are already rolled-out in large infrastructure projects, come under the suspicion of conspiracy and loose trust in the security community (and even worse in the user community).

Flexibility, Agility, and Costs

In the discussion on new curves within the CFRG, some of the proponents of special curves argue that a single set of curves is sufficient for all use cases. We do not share this opinion for several reasons. First, we consider at least two set of curves as necessary: one for high speed in software and one for high-assurance applications. Nevertheless, those two worlds have to be able to interact. Since we already argued that special primes are not suited for high-assurance solutions, we assume the usage of a verifiably pseudo-random curve for this communication. Second, a single curve would be an exposed target for an adversary. Using several different parameters (and even changing on a regular basis) might prevent an adversary from launching an attack in the first place due to the reduced gain[5]. Finally, even though we currently expect only generic attacks on curves, we cannot be sure about future developments.

The costs for implementation and usage of the curves depend not only on their implementation and runtime. In particular, for high-assurance devices, the costs for evaluation and certification have to be considered as well. Further, it has to be noted that for server implementations, i.e., where high-performance curves are preferred, the cost of supporting additional curves depends on the question how often the additional curves are used – supporting two different sets of curves does not affect the performance of special-prime curves in general.

---

[4] Some implementations in the field neglected this check for years. This made RFC 6989 necessary.

[5] For example, attacks using Time-Memory-Trade-offs could become attractive if the adversary is sure that only one curve will be used for a large period of time.

Interoperability

The discussion on rigidity, fast implementations and different use cases have led to a trend in which many groups created their own parameter set. Hence, there is a huge variety of different currently used curves, impeding interoperability. Further, this increases implementation costs and leads to irritation of potential users. We see this as a significant obstacle to the success of ECC. In order to overcome this obstacle, we believe a concerted effort of all potential interest groups should generate an agreed small set of curves that is commonly promoted.

Patents

Elliptic curves are often considered a minefield when it comes to patents. The situation for pseudo-random primes seems, due to their current wide-spread usage, to be rather clear, while the question which patents apply to special optimizations is still opaque.

Conclusion

In the current discussion of the CFRG, there is a prevalence of requirements for software implementations in secure environments, in particular, the demand for optimized arithmetic and special prime structures. If nothing else, the focus of a majority of ECC Brainpool members on hardware and high-assurance software is leading to other needs, i.e., flexibility and security is most important and performance ranks third. We think that the previously mentioned considerations and conclusions provide a different and necessary viewpoint to the current discussion on the development and the selection of parameters for future elliptic curves. A potential compromise is the construction of new curves with similar properties as the Brainpool Curves, i.e., with verifiably pseudo-random primes, and make this set mandatory to implement. Thus, we would have two sets of curves, one for high-speed software in secure environments and one for hardware and with lower speed in software (for hostile environments).

Meeting Participants

Besides the authors, the following people have participated at the ECC Brainpool meeting. Not all aspects were unanimously agreed upon in all detail. However, the reflections on special primes and implementation security (for hardware) found great endorsement. All see the great risk that securely chosen curves, as the current Brainpool curves, come under the suspicion of conspiracy and loose trust in the security and user community.

Jörg Albrecht, Utimaco, joerg.albrecht<at>utimaco.com

Dr. Dirk Feldhusen, SRC, dirk.feldhusen<at>src-gmbh.de

Dr. Wieland Fischer, Infineon, wieland.fischer<at>infineon.com

Dr. Guido Frank, BSI, guido.frank<at>bsi.bund.de

Prof. Dr. Gerhard Frey, Universität Duisburg-Essen, frey<at>exp-math.uni-essen.de

Prof. Dr. Joachim von zur Gathen, BIT Universität Bonn, gathen<at>bit.uni-bonn.de

Prof. Dr. Ernst-Günter Giessmann, T-Systems, ernstg.giessmann<at>t-systems.com

Martin Goldack, TÜViT, m.goldack<at>tuvit.de

Alexander Goth, Universität Bonn, alexandergoth<at>yahoo.de

Andreas Hallof, gematik, andreas.hallof<at>gematik.de

Dr. Erwin Hess, Siemens, erwin.hess<at>siemens.com

Dr. Thomas Hesselmann, BSI , thomas.hesselmann<at>bsi.bund.de

Gesine Hinterwälder, Universität Bochum, gesine.hinterwaelder<at>rub.de

Dr. Stavros Kousidis, BSI, stavros.kousidis<at>bsi.bund.de

Dr. Dennis Kügler, BSI, dennis.kuegler<at>bsi.bund.de

Sebastian Kutzner, TÜViT, s.kutzner<at>tuvit.de

Dr. Daniel Loebenberger, BIT Universität Bonn, daniel<at>bit.uni-bonn.de

Michael Müller, Rohde & Schwarz SIT, michael-d.mueller<at>rohde-schwarz.com

Dr. Kim Nguyen, Bundesdruckerei, kim.nguyen<at>bdr.de

Dr. Gesa Ott, Utimaco, gesa.ott<at>utimaco.com

Dr. Matthias Peter, BSI, matthias.peter<at>bsi.bund.de

Dr. Susanne Pingel, BSI, susanne.pingel<at>bsi.bund.de

Dr. Martin Seysen, Giesecke & Devrient, martin.seysen<at>gi-de.com

Sebastian Stappert, NXP, sebastian.stappert<at>nxp.com

Prof. Dr. Oliver Stein, OTH Regensburg, oliver.stein<at>oth-regensburg.de

Annika Strobel, escrypt, annika.strobel<at>escrypt.com

Olaf Stücker, T-Systems, olaf.stuecker<at>t-systems.com

Dr. Jens Tölle, Fraunhofer FKIE, jens.toelle<at>fkie.fraunhofer.de

Guntram Wicke, T-Systems, guntram.wicke<at>t-systems.com

Dr. Andreas Wiemers, BSI, andreas.wiemers<at>bsi.bund.de

Dr. Thomas Zeggel, Cryptovision, thomas.zeggel<at>cryptovision.com

Ralf Zimmermann, Universität Bochum, zimmermann<at>crypto.rub.de

References

[1] K. Paterson: “Formal request from TLS WG to CFRG for new elliptic curves.” CFRG mailing list. July, 14, 2014. http://www.ietf.org/mail-archive/web/cfrg/current/msg04655.html.

[2] Bundesamt für Sicherheit in der Informationstechnik: “Minimum Requirements for Evaluating Side-Channel Attack Resistance of Elliptic Curve Implementations.” Version 1.04, 01.07.2011.

[3] M. Lochter and J. Merkle: Elliptic Curve Cryptography (ECC) Standard Curves and Curve Generation. RFC 5639, March 2010.

[4] D.J. Bernstein: "Curve25519: New Diffie-Hellman Speed Records." Pages 207–228 in: Public Key Cryptography — PKC 2006. Lecture Notes in Computer Science 3958, Springer, 2006.

[5] B.E. Black, J. Bos, C. Costello, P. Longa, and M. Naehrig: "Elliptic Curve Cryptography (ECC) Nothing Up My Sleeve (NUMS) Curves and Curve Generation", June 2014. http://www.ietf.org/id/draft-black-numscurves-01.txt.

[6] D.J. Bernstein, T. Chou, C. Chuengsatiansup, A. Huelsing, T. Lange, R. Niederhagen, and C. van Vredendaal: “How to manipulate curve standards: a white paper for the black hat”, Cryptology ePrint Archive, Report 2014/571.