Compact implementations of pairings

Anthony Van Herrewege

***Abstract*—I present you the best ASIC implementation in the whole wide world.**

***Index Terms*—Cryptography, pairings, Tate, ASIC**

I. Inleiding

INSGELIJKS dit inleidende hoofdstuk zal enige achtergrond informatie verschaft worden omtrent cryptografie. Verder wordt het concept identiteitsgebaseerde cryptografie duidelijk gemaakt. Er zal uitgelegd worden waarom de recente ondekking van pairings hier zo belangrijk voor is. Ten slotte geeft een kort overzicht aan wat in de literatuur terug te vinden is qua implementaties van pairings. In het volgende hoofdstuk wordt de werking van pairings dan wiskundig uitgespit.

A. Basisachtergrond cryptografie

Sinds het begin der tijden is er een nood geweest aan manieren om berichten versleuteld te verzenden tussen twee partijen. Voorbeelden van enkele klassieke encryptiemethoden zijn het Atbashcijfer [1] (Babylonië, 600 v. Chr.), het Caesarcijfer [2] (Rome, 56 n. Chr.) en het dubbele transpositiecijfer [3] (o.a. gebruikt door weerstandsgroepen in WO II). Eén eigenschap die al deze methodes met elkaar gemeen hebben, is het gebruik van dezelfde sleutel voor versleutelen en ontcijferen. Ook vele moderne encryptiemethodes, zoals bijvoorbeeld 3DES [4] en AES [5], gebruiken dit principe, dat men symmetrische versleuteling noemt.

In Fig. 1 wordt de algemene werking van een symmetrische encryptie getoond. Alice zendt een bericht M naar Bob door het te versleutelen, vercijferd met een door hen beiden gekende sleutel k. Bob op zijn beurt ontcijfert met diezelfde sleutel de cijfertekst C. Indien Eve de vooraf afgesproken sleutel kent, kan zij alle communicatie tussen Alice en Bob ontcijferen. Er is dus nood aan een manier om veilig een sleutel k te kunnen afspreken tussen twee partijen.

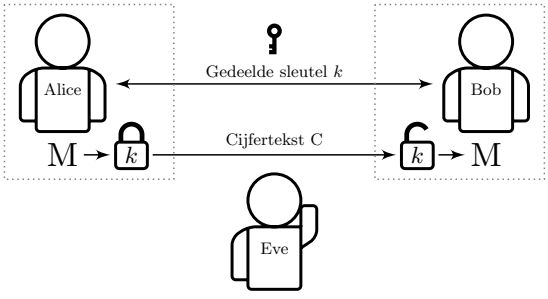

Figure 1. Algemene werking van symmetrische encryptie

Een oplossing voor het veilig afspreken van een gedeelde sleutel was tot 1976 niet gekend. Toen stelden Diffie en Hellman hun algoritme voor sleuteluitwisseling over een onbeveiligd kanaal voor [6]. Deze ontdekking plaveide de weg voor asymmetrische cryptografie (ook wel publieke sleutel cryptografie genoemd). Met behulp van dit type cryptografie kunnen eveneens boodschappen versleuteld worden. Dit wordt geïllustreerd in Fig. 2. Wanneer Alice een bericht M naar Bob wil versturen, zoekt ze eerst zijn publieke sleutel k_P op in een databank. Vervolgens versleutelt ze haar bericht met die publieke sleutel. Enkel Bob kan met behulp van zijn geheime sleutel k_G dan het bericht ontcijferen. Een systeem als dit biedt het grote voordeel dat er geen nood is om de gebruikte (publieke) sleutel geheim te houden. Het is namelijk onmogelijk om met de publieke sleutel de cijfertekst te ontcijferen. Eve heeft er in dit geval dus geen baat bij de gebruikte publieke sleutel te onderscheppen.

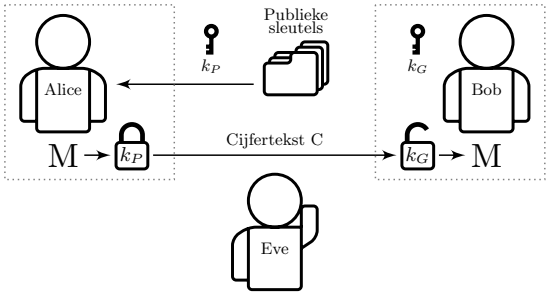

Figure 2. Algemene werking van asymmetrische encryptie

Een andere toepassing van asymmetrische cryptografie is het plaatsen en verifiëren van digitale handtekeningen. Digitale handtekeningen zijn vergelijkbaar met klassieke handtekeningen. Ze kunnen, indien juist geïmplementeerd, gebruikt worden om te verifiëren dat een bepaald bericht effectief door de persoon verstuurd is die de verzender beweert te zijn. Ook kan men aan de hand van een digitale handtekening nagaan of de inhoud van een bericht niet gewijzigd werd door een derde persoon tijdens de verzending. ECDSA [7] en RSA [8] zijn enkele van de vele cryptografische algoritmes die toelaten digitale handtekeningen te genereren.

Om te verzekeren dat de publieke sleutels van elke mogelijke ontvanger voorradig zijn, dient een soort een centrale databank voorzien te worden. Indien iemand het voor anderen mogelijk wil maken hem versleutelde berichten te versturen, genereert die persoon eerst een publieke en private sleutel. De publieke sleutel wordt vervolgens naar de database gestuurd, waar iedereen hem kan ophalen.

B. Identiteitsgebaseerde cryptografie

Een nadeel van de publieke sleutel cryptografie zoals voorgesteld in de vorige paragraaf zit hem in het sleutelbeheer. Er is geen manier om zeker te zijn dat wanneer de publieke

Anthony Van Herrewege can be contacted at anthonyvh@gmail.com.

sleutel van Bob opgevraagd wordt de verkregen sleutel effectief die van Bob is. Indien Eve bijvoorbeeld haar publieke sleutel in de database laat opslaan onder Bobs naam, zal Alice berichten voor Bob versleutelen met Eves publieke sleutel. Een mogelijke oplossing hiervoor is bijvoorbeeld het "web of trust", zoals geïmplementeerd door de software PGP [9]. Daarbij kunnen mensen aangeven of ze een bepaalde publieke sleutel betrouwbaar vinden of niet. Een sleutel die vergezeld wordt van meerdere getuigenissen van betrouwbaarheid zal dat dan waarschijnlijk ook zijn. Verder bestaat er een zogenaamde "revocation list", die aangeeft welke sleutels niet meer geldig zijn.

Uiteraard is ook zo een systeem niet volledig waterdicht. Iemand kan bijvoorbeeld onder verschillende identiteiten sleutels insturen en vervolgens met al die verschillende identiteiten zijn sleutels een certificaat van vertrouwen geven. Indien iemands publieke sleutel zou afgeleid kunnen worden van bekende gegevens omtrent zijn identiteit dan zouden deze problemen niet bestaan.

In 1984 stelde Shamir een methode voor waarbij dit mogelijk zou zijn [10]. Het basisidee is als volgt: in plaats van een centrale databank voor publieke sleutels is er een centrale server die private sleutels voor elke gebruiker genereert aan de hand van geheime parameters. Gebruikers kunnen hun private sleutel dus niet zelf berekenen. De centrale server publiceert ook informatie omtrent hoe iemands identiteitsgegevens kunnen worden omgezet naar een publieke sleutel. Wanneer een gebruiker wil deelnemen aan beveiligde communicatie, meldt hij zich aan bij de centrale server en verkrijgt hij zijn private sleutel alsook de parameters om publieke sleutels te berekenen. Voor zowel de private sleutel als de parameters wordt er van uit gegaan dat deze levenslang gelden. Een gebruiker dient zich dus slechts éénmalig aan te melden.

Uiteraard is ook dit concept niet zonder problemen. Indien bijvoorbeeld de geheime parameters van de centrale server achterhaald worden, is het onmogelijk gebruikers daarvan op de hoogte te brengen. Een mogelijke oplossing is elke gebruiker om de zoveel tijd te voorzien van een nieuwe geheime sleutel. In dat geval stelt zich echter een nieuw probleem, want dan moet een methode bedacht worden om alle nieuwe sleutels bij de gebruikers te krijgen.

Door deze problemen is de toepassing van identiteitsgebaseerde cryptografie eerder geschikt voor kleine groepen mensen, bijvoorbeeld intern in een bedrijf. In dat geval kost het weinig moeite iedereen op regelmatige tijdstippen van nieuwe sleutels te voorzien.

Een andere ideale toepassing is het gebruik van dit type cryptografie in netwerken van sensoren. Zo'n netwerk kan bestaan uit honderden nodes met een beperkte reken- en vermogenscapaciteit. Indien publieke sleutel cryptografie als vanouds zou worden gebruikt, zou dit leiden tot veel extra communicatie tussen de nodes en een centrale server. Telkens de nodes meetgegevens naar de server willen sturen, zouden ze diezelfde server eerst moeten contacteren om zijn publieke sleutel te weten te komen. In het omgekeerde geval zou de server telkens hij een node wil contacteren hetzelfde moeten doen. Als echter identiteitsgebaseerde cryptografie wordt toegepast, is al die extra communicatie niet meer nodig, aangezien men de benodigde publieke sleutels kan berekenen met behulp van een unieke ID.

Een uitvoerig overzicht van identiteitsgebaseerde cryptografie en de bijhorende mogelijkheden en problemen valt buiten het bestek van thesis. In het volgende hoofdstuk zullen wel enkele mogelijkheden naderbij bestudeerd worden. Een uitstekend startpunt voor meer informatie is [11].

C. Pairings

Hoewel het idee reeds in 1984 gepubliceerd werd, zou het echter tot 2001 duren eer Boneh en Franklin een efficiënt algoritme voor identiteitsgebaseerde cryptografie voorstelden [12]. Zij stelden een schema op dat toeliet de ideeën van Shamir ook effectief te implementeren. In hun voorstel maakten ze gebruik van de Weil pairing [13]. Al gauw verschenen er variaties op het oorspronkelijke schema. Daarin werd het gebruik van andere pairings voorgesteld, zoals bijvoorbeeld de Tate [14] of de η_T [15] pairing. Wat pairings juist zijn en hoe ze gebruikt kunnen worden, zal in het volgende hoofdstuk uitvoerig aan bod komen.

Alvorens de wiskunde achter pairings in te duiken, wordt eerst nog een overzicht gegeven van de huidige "state of the art" van implementaties van pairings. De mogelijkheden van implementaties op een microchip en een FPGA passeren de revue. Implementaties van pairings op computers zijn per definitie niet in een omgeving met beperkte ressources toepasbaar. Ze hebben dus weinig te maken met deze thesis, die als uitgangspunt compacte implementaties heeft. Vandaar dat dit type implementaties dan ook niet bestudeerd zal worden.

1) Microchip implementaties: Implementaties van pairings voor gebruik op een MICA node [16], specifiek ontwikkeld voor gebruik in netwerken van sensoren, worden voorgesteld in [17], [18] en [19]. De processor op deze node is een ATMega128L microchip [20]. Een overzicht van de resultaten is gegeven in Table I. Rekening houdend met het stroomverbruik en de batterijspanning gegeven in [19], is het vermogenverbruik waarschijnlijk ongeveer $23.60\ mW$.

Table I
RESULTATEN UIT DE LITERATUUR VOOR IMPLEMENTATIES ONTWIKKELD OP EEN MICA NODE [16]

	TinyTate [17]	TinyPBC [18]	NanoECC [19]	
			Binair	Priem
Veld	$\mathbb{F}_p$ 256 bit	$\mathbb{F}_{2^{271}}$	$\mathbb{F}_{2^{163}}$	$\mathbb{F}_p$ 160 bit
Pairing	Tate	η_T	Tate	Tate
Rekentijd (s)	30.21	5.45	10.96	17.93

2) FPGA implementaties: In de literatuur zijn vrij veel ontwerpen voor FPGA's terug te vinden. Het probleem is echter dat men zich bij het ontwerp hiervan steeds toelegt op het behalen van een zo hoog mogelijke snelheid, wat resulteert in een grote oppervlakte. Dit type implementaties is dus minder geschikt zijn voor toepassingen met beperkte ressources.

Toch wordt in Table II een sumier overzicht gegeven van een zeer beperkt aantal ontwerpen. Bij de selectie hiervan werd vooral gekozen voor ontwerpen waarin in een vrij klein veld

gerekend werd. Er dient in acht te worden genomen dat bij al deze implementaties snelheid, en niet een compacte, zuinige implementatie, het voornaamste doel is.

Table II
RESULTATEN UIT DE LITERATUUR VOOR IMPLEMENTATIES ONTWIKKELD VOOR FPGA'S

	Veld	Pairing	Opp. [slices]	f [MHz]	Reken-tijd [μs]
Ronan *et al.* [21]	$\mathbb{F}_{2^{103}}$	Tate	21021	51	206
Shu *et al.* [22]	$\mathbb{F}_{2^{239}}$	Tate	25287	84	41
Keller *et al.* [23]	$\mathbb{F}_{2^{251}}$	Tate	27725	40	2370
Grabher en Page [24]	$\mathbb{F}_{3^{97}}$	Tate	4481	150	432
Beuchat *et al.* [25]	$\mathbb{F}_{3^{97}}$	η_T	1833	145	192

REFERENCES

[1] C. D. Isbell, "Some Cryptograms in the Aramaic Incantation Bowls," *Journal of Near Eastern Studies*, vol. 33, no. 4, pp. 405–407, 1974.

[2] C. A. Deavours, D. Kahn, L. Kruh, G. Mellen, and B. Winkel, Eds., *Cryptology: yesterday, today, and tomorrow*. Norwood, MA, USA: Artech House, Inc., 1987.

[3] D. Kahn, *The Codebreakers: The Story of Secret Writing*. New York: The Macmillan Company, 1967.

[4] W. C. Barker, *Recommendation for the Triple Data Encryption Algorithm (TDEA) block cipher*. National Institute for Standards and Technology, 2004 .

[5] J. Daemen and V. Rijmen, *The Design of Rijndael: AES – The Advanced Encryption Standard*. Springer-Verlag, 2002.

[6] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *Transaction on Information Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.

[7] D. R. L. Brown, "Generic Groups, Collision Resistance, and ECDSA," *Des. Codes Cryptography*, vol. 35, no. 1, pp. 119–152, 2005.

[8] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[9] P. Zimmermann, *PGP source code and internals*. Cambridge, MA, USA: MIT Press, 1995.

[10] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of CRYPTO 84 on Advances in cryptology*. New York, NY, USA: Springer-Verlag New York, Inc., 1985, pp. 47–53.

[11] M. Maas, "Pairing-Based Cryptography," Master's thesis, Technische Universiteit Eindhoven, januari 2004.

[12] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[13] V. S. Miller, "The Weil Pairing, and Its Efficient Calculation," *Journal of Cryptology*, vol. 17, no. 4, pp. 235–261, 2004.

[14] G. Frey, M. Muller, and H. Rück, "The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystem," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1717–1719, 1999.

[15] P. S. L. M. Barreto, S. D. Galbraith, C. ÓhÉigeartaigh, and M. Scott, "Efficient pairing computation on supersingular Abelian varieties." *Des. Codes Cryptography*, vol. 42, no. 3, pp. 239–271, 2007.

[16] J. L. Hill and D. E. Culler, "Mica: a wireless platform for deeply embedded networks," *Micro, IEEE*, vol. 22, no. 6, pp. 12–24, 2002.

[17] L. B. Oliveira, D. F. Aranha, E. Morais, F. Daguano, J. López, and R. Dahab, "TinyTate: Computing the Tate Pairing in Resource-Constrained Sensor Nodes," in *Sixth IEEE International Symposium on Network Computing and Applications*. IEEE Computer Society, 2007, pp. 318–323.

[18] L. B. Oliveira, M. Scott, J. López, and R. Dahab, "TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks," in *5th International Conference on Networked Sensing Systems*, 2008, pp. 173–180.

[19] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab, "NanoECC: Testing the Limits of Elliptic Curve Cryptography in Sensor Networks." *European conference on Wireless Sensor Networks (EWSN08)*, 2008.

[20] Atmel, *ATMega128 datasheet*. [Online]. Available: http://www.atmel.com

[21] R. Ronan, C. ÓhÉigeartaigh, C. C. Murphy, M. Scott, and T. Kerins, "Hardware acceleration of the Tate pairing on a genus 2 hyperelliptic curve," *Journal of Systems Architecture*, vol. 53, no. 2-3, pp. 85–98, 2007.

[22] C. S. Soonhak, C. Shu, S. Kwon, and K. Gaj, "FPGA Accelerated Tate Pairing Based Cryptosystems over Binary Fields," in *Cryptology ePrint Archive, Report 2006/179*, 2006.

[23] M. Keller, T. Kerins, F. M. Crowe, and W. P. Marnane, "FPGA Implementation of a GF(2^m) Tate Pairing Architecture." in *ARC*, ser. Lecture Notes in Computer Science, K. Bertels, J. a. M. P. Cardoso, and S. Vassiliadis, Eds., vol. 3985. Springer, 2006, pp. 358–369.

[24] P. Grabher and D. Page, "Hardware Acceleration of the Tate Pairing in Characteristic Three." in *CHES*, ser. Lecture Notes in Computer Science, vol. 3659. Springer, 2005, pp. 398–411.

[25] J.-L. Beuchat, N. Brisebarre, J. Detrey, E. Okamoto, M. Shirase, and T. Takagi, "Algorithms and Arithmetic Operators for Computing the η_T Pairing in Characteristic Three." *IEEE Trans. Computers*, vol. 57, no. 11, pp. 1454–1468, 2008.

Anthony Van Herrewege Testen maar Nog eentje