(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 101 61 137 B4** 2008.02.14

(12) # Patentschrift

(21) Aktenzeichen: **101 61 137.4**
(22) Anmeldetag: **12.12.2001**
(43) Offenlegungstag: **02.10.2003**
(45) Veröffentlichungstag
der Patenterteilung: **14.02.2008**

(51) Int Cl.[8]: ***H04L 9/30*** (2006.01)

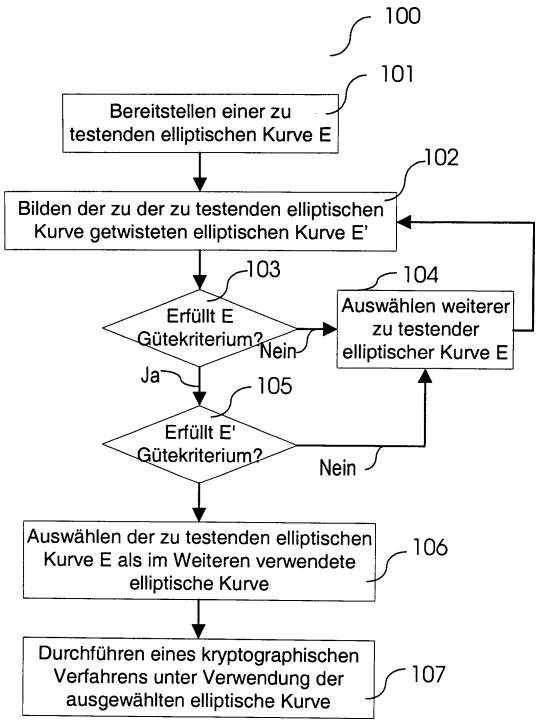
(73) Patentinhaber:
Siemens AG, 80333 München, DE

(72) Erfinder:
Hess, Erwin, Dr., 85521 Ottobrunn, DE; Meyer, Bernd, Dr., 81737 München, DE

(56) Für die Beurteilung der Patentfähigkeit in Betracht gezogene Druckschriften:
US 61 41 420 A
US 58 05 703 A
WO 99/49 386 A1

(54) Bezeichnung: **Verfahren und System zum kryptographischen Bearbeiten von Daten**

(57) Hauptanspruch: Verfahren zum rechnergestützten Erzeugen und Verifizieren einer digitalen Signatur,
– bei dem von einem ersten Rechner folgende Verfahrensschritte durchgeführt werden:
– eine elliptische Kurve über einem Körper der Charakteristik 2 wird bereitgestellt,
– mindestens ein Punkt, der auf der elliptischen Kurve liegt, wird ausgewählt oder ermittelt,
– nur eine Koordinate des Punktes wird gespeichert,
– Daten werden gemäß einem vorgegebenen Verfahren zum Erzeugen einer digitalen Signatur kryptographisch bearbeitet, wobei bei dem gesamten kryptographischen Verfahren nur die eine Koordinate des Punktes verwendet wird,
– bei dem die kryptographisch bearbeiteten Daten von dem ersten Rechner zu einem zweiten Rechner übertragen werden, wobei von auf der elliptischen Kurve sich befindenden Punkten, welche von dem ersten Rechner zu dem zweiten Rechner übertragen werden, jeweils nur die eine Koordinate des jeweiligen Punktes übertragen wird, und – bei dem von dem zweiten Rechner zumindest folgende Verfahrensschritte durchgeführt werden:...



Beschreibung

[0001] Die Erfindung betrifft ein Verfahren und eine Vorrichtung zum Ermitteln einer elliptischen Kurve sowie ein Verfahren und eine Vorrichtung zum Multiplizieren eines Punktes mit einem Skalar.

[0002] Kryptographische Verfahren auf Basis elliptischer Kurven, welche über endlichen Körpern definiert sind, werden immer häufiger im Rahmen sogenannter Public-Key-Verfahren (asymmetrischer Krypto-Verfahren) eingesetzt, d.h. im Rahmen kryptographischer Verfahren, welche zwei unterschiedliche Schlüssel, einen geheimen kryptographischen Schlüssel sowie einen öffentlichen nicht-geheimen Schlüssel verwenden.

[0003] Die Krypto-Verfahren auf Basis elliptischer Kurven sind sehr effizient, was insbesondere daran liegt, dass bei elliptischen Kurven im Gegensatz zu früheren Public-Key-Verfahren keine Angriffsmethoden mit subexponentieller Laufzeit bekannt sind.

[0004] Anders ausgedrückt bedeutet dies, dass der Sicherheitsgewinn pro Bit der verwendeten Sicherheitsparameter bei Verfahren auf Basis elliptischer Kurven höher ist und somit für praktische Anwendungen deutlich kürzere Schlüssellängen verwendet werden können.

[0005] Somit sind die kryptographischen Verfahren auf Basis elliptischer Kurven performanter und benötigen eine geringere Bandbreite zur Übertragung der Systemparameter als andere Public-Key-Verfahren bei vergleichbarem Grad an erreichbarer kryptographischer Sicherheit.

[0006] Eine elliptische Kurve E wird allgemein durch eine kubische Gleichung der folgenden Form beschrieben:

$$y^2 + a_1 \cdot x \cdot y + a_3 \cdot y = x^3 + a_2 \cdot x^2 + a_4 \cdot x + a_6, \quad (1)$$

wobei a_1, a_2, a_3, a_4, a_6 fixe Elemente eines endlichen Körpers K sind, welche die elliptische Kurve E parametrisieren.

[0007] In diesem Zusammenhang ist anzumerken, dass in Abhängigkeit von der Charakteristik des Körpers K die Kurvengleichung der elliptischen Kurve E auf einfachere Kurvengleichungen transformiert werden kann.

[0008] Die Menge aller Paare (x, y) aus K^2 welche die gegebene Kurvengleichung der elliptischen Kurve E erfüllen, werden im Weiteren als die Punkte P der elliptischen Kurve E bezeichnet. Auf der Menge der Punkte P einer elliptischen Kurve E kann eine abelsche Gruppenstruktur G definiert werden. Die Gruppenstruktur G induziert eine Skalarmultiplikation $Z \cdot G \rightarrow G$ von Kurvenpunkten mit ganzen Zahlen, die die Grundlage aller kryptographischen Verfahren auf Basis elliptischer Kurven bildet:
Sei s eine ganze Zahl, P ein Punkt der elliptischen Kurve E und $Q = s \cdot P$ das s-fache des Punktes P. Sind die Punkte P und Q gegeben, so bezeichnet man die Berechnung eines geeigneten Skalar s mit $Q = s \cdot P$ als das diskrete Logarithmus-Problem für elliptische Kurven. Bei geeigneter Wahl des endlichen Körpers K und der Parameter der elliptischen Kurve E ist es mit derzeitigen algorithmischen Mitteln nicht in akzeptabler Zeit möglich, das diskrete Logarithmusproblem zu lösen.

[0009] Wie oben erläutert wurde, ist ein Punkt P einer elliptischen Kurve E durch seine x-Koordinate und seine y-Koordinate gegeben. Aufgrund der Kurvengleichung der elliptischen Kurve E existieren zu einem x-Wert höchstens zwei unterschiedliche y-Werte y_1 und y_2, so dass die Punkte (x, y_1) und (x, y_2) Punkte auf der elliptischen Kurve E sind. Um somit einen Punkt auf der elliptischen Kurve E eindeutig festzulegen, ist außer der x-Koordinate nur noch ein Bit an zusätzlicher Information erforderlich.

[0010] In dem Fall einer elliptischen Kurve E über endlichen Primkörpern genügt beispielsweise das sogenannte Least Significant Bit (LSB) der y-Koordinate oder das Vorzeichen der y-Koordinate des jeweiligen Punktes als zusätzliche Information.

[0011] Um Kurvenpunkte, d.h. Punkte auf der elliptischen Kurve E kompakter zu speichern und effizienter übertragen zu können, ist es bekannt, sogenannte Punktkompressionsverfahren und Punktdekompressionsverfahren einzusetzen, bei denen die Y-Koordinate auf dieses zusätzliche Bit reduziert wird bzw. von dem reduzierten Bitwert der tatsächliche Wert der y-Koordinate des Punktes auf der elliptischen Kurve rekonstruiert wird.

[0012] Die bekannten kryptographischen Verfahren auf Basis elliptischer Kurven sind jedoch hinsichtlich sogenannter Seitenkanalangriffe verletzbar, wie z.B. der sog. Simple-Power-Analyse (vgl. [1]), der sogenannten Differential-Power-Analyse (vgl. [2]), der sogenannten Differential-Fault-Analyse (vgl. [3], [4]) sowie den sogenannten Timing-Angriffen (vgl. [5]).

[0013] Es ist erkannt worden, dass die konkrete Implementierungsweise von Krypto-Verfahren, welche auf elliptischen Kurven basieren, für den schlussendlich resultierenden Grad an erreichbarer Sicherheit der jeweiligen Applikationen von erheblicher Bedeutung ist (vgl. [6]).

[0014] Wie in [3] und in [4] beschrieben ist, wird bei der Differential-Fault-Analyse versucht, mittels Induzierens von Berechnungsfehlern oder mittels Stören der Eingangsdaten fehlerhafte Berechnungsergebnisse der kryptographischen Protokolle zu verursachen, um dann letztlich aus den fehlerhaften Ergebnissen auf geheime Parameter wie beispielsweise den verwendeten geheimen Schlüssel, d.h. den Private-Key, Rückschlüsse ziehen zu können, im günstigsten Fall diesen sogar vollständig zu ermitteln. Dabei versucht der Angreifer entweder, die von der Anwendung verwendete Punktgruppe der elliptischen Kurve zu verändern oder die Gruppenstruktur G vollständig zu zerstören. Diese Situation tritt hauptsächlich in den beiden folgenden Fällen auf:

– Punkte der elliptischen Kurve werden als Protokoll-Parameter von nicht vertrauenswürdigen Protokollpartnern übergeben, oder

– Punkte der elliptischen Kurve sind Ereignisse von möglicherweise gestörten Berechnungen der eigenen Anwendung.

[0015] Zur Abwehr der oben beschriebenen Seitenkanalangriffe ist es ferner bekannt, dass gegebene oder berechnete Punkte der elliptischen Kurve auf deren Korrektheit hin überprüft werden. Sind Punkte der elliptischen Kurve durch die jeweilige x-Koordinate und die y-Koordinate vollständig gegeben, so kann getestet werden, ob die Punkte

1. die Kurvengleichung der elliptischen Kurve E erfüllen, und
2. in der korrekten Untergruppe der elliptischen Kurve liegen, d.h., dass sie nicht in einer Untergruppe kleiner Ordnung enthalten sind.

[0016] Eine solche Überprüfung ist jedoch bei Punkten, welche gemäß einem Punktkompressionsverfahren bzw. einem Punktdekompressionsverfahren in komprimierter Form vorliegen, nicht unmittelbar durchführbar.

[0017] Ist zumindest einer der Koordinatenwerte des Punktes auf der elliptischen Kurve komprimiert, so ist zunächst aus der nicht-komprimierten, vorzugsweise der x-Koordinate und der jeweiligen Zusatzinformation die vollständige Koordinate des komprimierten Wertes, üblicher Weise die vollständige y-Koordinate zu errechnen. Diese Berechnung benötigt jedoch einen erheblichen zusätzlichen Rechenaufwand, da in diesem Zusammenhang eine quadratische Gleichung über einem endlichen Körper gelöst werden muss.

[0018] Ferner ist in [6] beschrieben, dass zur Abwehr eines auf einer Power-Analyse basierenden Seitenkanalangriffs die projektive Koordinatendarstellung einer elliptischen Kurve randomisiert wird.

[0019] Weiterhin ist in [7] ein Verfahren zur Transformation einer elliptischen Kurve beschrieben, welche über einem endlichen Körper der Charakteristik 2 definiert sind, wobei das Verfahren für diese spezielle elliptische Kurve es ermöglicht, ein skalares Vielfaches eines Punktes der jeweiligen elliptischen Kurve unter Verwendung der x-Koordinate zu berechnen.

[0020] Bei dieser Vorgehensweise, welche auch in [8] beschrieben ist, ist jedoch ein durchgängiger Verzicht auf die y-Koordinate nicht möglich. Bei diesem Verfahren muss nämlich die y-Koordinate eines Vielfachen des Basispunktes auf der elliptischen Kurve berechnet werden und nur der letzte Verifikationsschritt kommt ohne die jeweilige y-Koordinate aus.

[0021] Außerdem müssen bei diesen in [7] und in [8] beschriebenen Verfahren mehrere modulare Divisionen in Z/pZ durchgeführt werden.

[0022] Weiterhin ist in [9] ein Verfahren zum Erzeugen und zum Verifizieren von elektronischen Unterschriften auf Basis von diskreten Logarithmen in der Punktegruppe von elliptischen Kurven, welche über endlichen Körpern definiert sind, ohne Verwendung von y-Koordinaten ebenfalls für die in [7] beschriebenen Montgomery-Kurven, beschrieben.

[0023] Im Rahmen des Verifikationsverfahrens gemäß [9] werden quadratische Polynome verwendet. Als

Beispiele sind Signaturen nach ElGamal EC-DSA angegeben. Es ist anzumerken, dass in [9] Additionsformeln und Polynome nur für die Montgomery-Kurven, wie sie in [7] beschrieben sind, zur Verifikation von digitalen Signaturen angegeben sind.

[0024] In [10] sind allgemeine Additionsformeln für elliptische Kurven über endlichen Körpern der Charakteristik größer als 3 angegeben, wobei diese aus der Exponentiationsmethode von Montgomery, wie sie in [7] beschrieben ist, abgeleitet wurden.

[0025] Das Dokument WO 99/49386 A1 beschreibt ein Signatur-Verifikationsverfahren, bei dem nur eine x-Koordinate eines Punktes einer elliptischen Kurve übermittelt wird und getestet wird, ob die beiden Punkte der elliptischen Kurve, die die x-Koordinate aufweisen können, eine Signaturbedingung erfüllen.

[0026] Somit liegt der Erfindung das Problem zu Grunde, ein Verfahren sowie ein System zur kryptographischen Bearbeitung von Daten anzubieten, bei dem gegenüber dem Stand der Technik eine weitere Reduktion an Rechenzeitbedarf für die Bearbeitung der Daten sowie eine Reduktion an benötigter Datenrate zur Übertragung von Daten von einem ersten Rechner zu einem zweiten Rechner erreicht wird.

[0027] Das Problem wird durch das Verfahren und das System zum kryptographischen Bearbeiten von Daten mit den Merkmalen gemäß den unabhängigen Patentansprüchen gelöst.

[0028] Bei dem Verfahren zum rechnergestützten kryptographischen Bearbeiten von Daten wird eine elliptische Kurve bereitgestellt und es wird mindestens ein Punkt, der auf der elliptischen Kurve liegt ausgewählt oder ermittelt. Für den jeweils ausgewählten oder ermittelten Punkt wird nur eine Koordinate, vorzugsweise die x-Koordinate gespeichert. Die Daten werden gemäß einem vorgegebenen kryptographischen Verfahren krpytographisch bearbeitet, wobei im Rahmen des gesamten kryptographischen Verfahrens, anders ausgedrückt während der gesamten kryptographischen Bearbeitung der Daten in dem ersten Rechner, nur jeweils die eine Koordinate, vorzugsweise somit nur die x-Koordinate des jeweiligen Punktes beziehungsweise der Punkte, die im Rahmen der kryptographischen Bearbeitung jeweils als Zwischenergebnisse oder als Ergebnisse berechnet werden, verwendet wird bzw. werden.

[0029] Die auf die oben beschriebene Weise bearbeiteten kryptographischen Daten werden von dem ersten Rechner zu einem zweiten Rechner, mit welchem der erste Rechner über ein Telekommunikationsnetz gekoppelt ist, übertragen.

[0030] Bei der Übertragung wird für alle übertragenen, sich auf der elliptischen Kurve befindenden Punkte jeweils nur die eine Koordinate des jeweiligen Punktes übertragen, vorzugsweise somit nur die x-Koordinate des jeweiligen Punktes.

[0031] Von dem zweiten Rechner werden die empfangenen, anders ausgedrückt die übertragenen kryptographisch bearbeiteten Daten gemäß einem weiteren vorgegebenen, vorzugsweise kryptographischen, Verfahren bearbeitet, allgemein werden die empfangen Daten zusätzlich erneut, vorzugsweise kryptographisch bearbeitet, wobei im Rahmen der gesamten empfangsseitigen Bearbeitung der empfangenen Daten wiederum nur die eine Koordinate eines jeweiligen Punktes auf der elliptischen Kurve verwendet wird, vorzugsweise somit nur die x-Koordinate des jeweiligen Punktes auf der elliptischen Kurve.

[0032] Ein System zur kryptographischen Bearbeitung von Daten, weist einen ersten Rechner und einen zweiten Rechner auf, welche über ein Telekommunikationsnetz miteinander gekoppelt sind. Die beiden Rechner weisen jeweils eine Prozessoreinheit auf die jeweils derart eingerichtet ist, dass sie die oben beschriebenen Verfahrensschritte in dem ersten Rechner beziehungsweise in dem zweiten Rechner durchführen können.

[0033] Anschaulich kann die Erfindung darin gesehen werden, dass durchgängig in der gesamten kryptographischen Bearbeitung von Daten senderseitig und empfängerseitig nur jeweils eine Koordinate eines Punktes auf der elliptischen Kurve verwendet wird, vorzugsweise nur die x-Koordinate.

[0034] Erfindungsgemäß wird somit eine erhebliche Reduktion im Rahmen des Rechenzeitbedarfs bei der kryptographischen Bearbeitung sowie auch bei der benötigten Datenrate im Rahmen der Übertragung von kryptographisch bearbeiteten Daten zwischen zwei miteinander gekoppelten Rechnern erreicht.

[0035] Anders ausgedrückt kann insbesondere auf die Speicherung, Übertragung und Bearbeitung von y-Koordinaten von Kurvenpunkten auf der elliptischen Kurve während der gesamten kryptographischen Bearbei-

tung der Daten und sogar der Übertragung der Daten verzichtet werden.

[0036] Es ist anzumerken, dass aus protokolltechnischer Sicht keine Einschränkungen für die jeweils einzusetzenden endlichen Körper, auf welchen die erfindungsgemäßen kryptographischen Verfahren basieren, existieren.

[0037] Bevorzugte Weiterbildung der Erfindungen ergeben sich aus den abhängigen Ansprüchen.

[0038] Das vorgegebene kryptographische Verfahren kann beispielsweise das Erzeugen digitaler Signaturen sein, in welchem Fall das weitere kryptographische Verfahren die Verifizierung einer erzeugten digitalen Signatur ist.

[0039] Alternativ kann das kryptographische Verfahren und das weitere, vorzugsweise kryptographische, Verfahren insgesamt ein Verfahren zum Schlüsselaustausch beziehungsweise zur Schlüsseleinigung zwischen zwei Teilnehmern einer Kommunikationsverbindung auf einen gemeinsamen Sitzungsschlüssel bilden, welcher zwischen dem ersten Rechner und dem zweiten Rechner im Rahmen einer nachfolgenden kryptographisch gesicherten Kommunikationsverbindung verwendet wird.

[0040] Gemäß einer anderen Ausgestaltung der Erfindung ist es vorgesehen, dass das kryptographische Verfahren ein Verfahren zum Verschlüsseln von Daten ist, und dass zur zusätzlichen kryptographischen Bearbeitung ein Verfahren zum Entschlüsseln von verschlüsselten Daten durchgeführt wird.

[0041] Gemäß einer anderen Ausgestaltung der Erfindung ist es vorgesehen, dass das kryptographische Verfahren und das zusätzliche kryptographische Verfahren insgesamt ein Verfahren zur Authentifikation des ersten Rechners bei dem zweiten Rechner bildet.

[0042] Ist das kryptographische Verfahren und das zusätzliche kryptographische Verfahren, d.h. die zusätzliche kryptographische Bearbeitung, insgesamt ein Verfahren zur Schlüsseleinigung auf einen kryptographischen Schlüssel, der von dem ersten Rechner und dem zweiten Rechner verwendet wird, so können die Verfahren auf einem Schlüsseleinigungsverfahren gemäß dem Prinzip gemäß Diffie-Hellman basieren.

[0043] Alternativ können die Verfahren auf einem sogenannten MTI-Protokoll basieren, insbesondere auf einem der folgenden MTI-Protokolle:

- dem MTI/A0-Protokoll,
- dem MTI/B0-Protokoll,
- dem MTI/C0-Protokoll, oder
- dem MTI/C1-Protokoll.

[0044] Gemäß der Ausgestaltung der Erfindung, bei dem die Verfahren ein Verfahren zum Verschlüsseln bzw. zum Entschlüsseln von Daten sind, ist gemäß einer Ausgestaltung der Erfindung vorgesehen, dass diese Verfahren auf dem Prinzip gemäß EC-ElGamal basieren.

[0045] Ist das kryptographische Verfahren ein Verfahren zum Erzeugen einer digitalen Signatur und das zusätzliche Verfahren ein Verfahren zum Überprüfen einer digitalen Signatur, so ist gemäß einer Ausgestaltung der Erfindung vorgesehen, dass die Signaturverfahren Verfahren sind, welche auf einem der folgenden Verfahren basieren:

- einem Verfahren zum Erzeugen und zum Prüfen von digitalen Signaturen gemäß EC-ElGamal,
- Verfahren zum Erzeugen und Prüfen von Signaturen gemäß EC-DSA, oder
- Verfahren zum Erzeugen von Signaturen und zum Prüfen von Signaturen gemäß EC-GDSA.

[0046] Gemäß einer anderen Weiterbildung der Erfindung ist es bei der Prüfung digitaler Signaturen auf Basis elliptischer Kurven vorgesehen einmal oder mehrmals ein oder mehrere Polynome auszuwerten, die nur von jeweils einer Koordinate eines Punktes oder der verwendeten Punkte abhängen.

[0047] Ausführungsbeispiele der Erfindung sind in den Figuren dargestellt und werden im Weiteren näher erläutert. Gleiche Elemente sind in den Ausführungsbeispielen teilweise mit gleichen Bezugszeichen versehen.

[0048] Auch wenn in den folgenden Ausführungsbeispielen jeweils bestimmte elliptische Kurven verwendet werden, ist die Erfindung jedoch nicht auf diese bestimmten elliptischen Kurven beschränkt, sondern sie kann für jede Art von elliptischer Kurve eingesetzt werden.

[0049] Es zeigen

[0050] Fig. 1 ein Ablaufdiagramm, in dem die einzelnen Schritte zum Ermitteln einer elliptischen Kurve gemäß einem Ausführungsbeispiel der Erfindung dargestellt sind;

[0051] Fig. 2 ein Blockdiagramm, in dem ein Verfahren zur Schlüsseleinigung gemäß einer Variante gemäß Diffie-Hellman unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist;

[0052] Fig. 3 ein Blockdiagramm, in dem eine erfindungsgemäße Variante des MTI/A0-Protokolls unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist;

[0053] Fig. 4 ein Blockdiagramm, in dem eine erfindungsgemäße Variante des MTI/B0-Protokolls unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist;

[0054] Fig. 5 ein Blockdiagramm, in dem eine erfindungsgemäße Variante des MTI/C0-Protokolls unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist;

[0055] Fig. 6 ein Blockdiagramm, in dem eine erfindungsgemäße Variante des MTI/C1-Protokolls unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist;

[0056] Fig. 7 ein Blockdiagramm, in dem die Verschlüsselung, Übertragung und Entschlüsselung von Nachrichten gemäß einer erfindungsgemäßen Variante gemäß ElGamal unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist;

[0057] Fig. 8 ein Blockdiagramm, in dem die Authentisierung mittels Public-Key-Verfahren unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist;

[0058] Fig. 9 ein Ablaufdiagramm, in dem eine erfindungsgemäße Variante des Erzeugens von ElGamal-Signaturen unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist;

[0059] Fig. 10 ein Ablaufdiagramm, in dem eine erfindungsgemäße Variante des Prüfens von ElGamal-Signaturen unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist;

[0060] Fig. 11 ein Ablaufdiagramm, in dem eine erfindungsgemäße Variante des Erzeugens von EC-DSA-Signaturen unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist;

[0061] Fig. 12 ein Ablaufdiagramm, in dem eine erfindungsgemäße Variante des Prüfens von EC-DSA-Signaturen unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist;

[0062] Fig. 13 ein Ablaufdiagramm, in dem eine erfindungsgemäße Variante des Erzeugens von EC-GDSA-Signaturen unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist; und

[0063] Fig. 14 ein Ablaufdiagramm, in dem eine erfindungsgemäße Variante des Prüfens von EC-GDSA-Signaturen unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve dargestellt ist.

[0064] Fig. 1 zeigt in einem Ablaufdiagramm **100** die einzelnen Verfahrensschritte zum rechnergestützten Ermitteln einer elliptischen Kurve gemäß einem Ausführungsbeispiel der Erfindung.

[0065] Das Verfahren wird von einem nicht dargestellten Rechner mit einem Prozessor und einem Speicher, in dem das Programmelement zur Durchführung der einzelnen weiteren beschriebenen Verfahrensschritte gespeichert ist, durchgeführt.

[0066] Der Rechner weist ferner eine Kommunikations-Eingangs-/Ausgangsschnittstelle auf, über die der Rechner mit anderen Rechnern oder anderen Kommunikationseinheiten Daten austauschen kann. Der Speicher und der Prozessor sind über einen Computerbus miteinander gekoppelt.

[0067] Gemäß diesem Ausführungsbeispiel der Erfindung ist in dem Speicher des Rechners eine Vielzahl von elliptischen Kurven gespeichert, oder es wird von dem Rechner eine Vielzahl von elliptischen Kurven gebildet.

[0068] Im Weiteren wird das erfindungsgemäße Verfahren anhand nur einer elliptischen Kurve aus der Menge

gespeicherter oder gebildeter elliptischer Kurven erläutert.

[0069] Erfindungsgemäß ist jedoch vorgesehen, das Verfahren beliebig oft für eine beliebige Anzahl der gespeicherten elliptischen Kurven durchzuführen und somit einen Teil oder alle der gespeicherten elliptischen Kurven dahingehend zu testen, ob sie einen ausreichenden Grad an kryptographischer Sicherheit bereitstellen, damit sie geeignet sind, für die im Weiteren beschriebenen kryptographischen Verfahren eingesetzt zu werden.

[0070] In einem ersten Schritt wird eine elliptische Kurve aus der Menge gespeicherter elliptischer Kurven als eine zu testende elliptische Kurve ausgewählt und somit bereitgestellt (Schritt **101**).

[0071] Elliptische Kurven über endlichen Körpern treten in Paaren auf. Zu einer elliptischen Kurve E(a_1, a_2, a_3, a_4, a_6), wobei mit a_1, a_2, a_3, a_4, a_6 jeweils Formparameter der elliptischen Kurve E bezeichnet werden, gehören sogenannte getwistete elliptische Kurven E'(b_1, b_2, b_3, b_4, b_6).

[0072] Die Punktezahlen der elliptischen Kurve E und der zu der elliptischen Kurve zugehörigen getwisteten elliptischen Kurve E' hängen voneinander ab und erfüllen folgende Gleichung:

$$\mathrm{ord}(E) + \mathrm{ord}(E') = 2\cdot|K| + 2. \quad (1)$$

[0073] Zu der ausgewählten zu testenden elliptischen Kurve E wird somit in einem weiterem Schritt (Schritt **102**) eine zu der elliptischen Kurve E zugehörige getwistete elliptische Kurve gebildet.

[0074] In einem weiteren Schritt (Schritt **103**) wird für die ausgewählte elliptische Kurve E geprüft, ob diese einem vorgegebenen kryptographischen Gütekriterium genügt (Schritt **103**), wobei gemäß diesem bevorzugten Ausführungsbeispiel das Gütekriterium dann erfüllt ist, wenn die Punktezahl ord(E) der elliptischen Kurve E einen Primteiler aufweist, der größer ist als ein vorgegebener Schwellenwert, die elliptische Kurve E weder supersingulär noch anomal ist und die Punktezahl ord(E) der elliptischen Kurve sowohl dem Frey-Rück-Kriterium als auch dem Menezes-Okamoto-Vanstone-Kriterium genügt.

[0075] Ist das Gütekriterium für die ausgewählte zu testende elliptische Kurve E nicht erfüllt, so wird in einem weiteren Schritt (Schritt **104**) eine weitere zu testende elliptische Kurve aus der Menge der gespeicherten elliptischen Kurven ausgewählt und das Verfahren wird ab Schritt **102** wiederholt.

[0076] Ist jedoch das Gütekriterium für die zu testende elliptische Kurve erfüllt, so wird in einem zusätzlichen Schritt (Schritt **105**) geprüft, ob auch die zu der elliptischen Kurve E zugehörige getwistete elliptische Kurve E' dem oben genannten kryptographischen Gütekriterium genügt.

[0077] Es ist in diesem Zusammenhang anzumerken, dass erfindungsgemäß beide Gütekriterien gleich sein können oder unterschiedliche Gütekriterien zur Bewertung der kryptographischen Güte der jeweiligen elliptischen Kurve bzw. getwisteten elliptischen Kurve verwendet werden können.

[0078] Anders ausgedrückt bedeutet dies, dass gemäß diesem Ausführungsbeispiel auch für die getwistete elliptische Kurve E' geprüft wird, ob die Punktezahl ord(E') der getwisteten elliptischen Kurve E' einen Primteiler besitzt, der größer ist als ein vorgegebener Schwellenwert, wobei der Schwellenwert gleich oder unterschiedlich zu dem Schwellenwert im Rahmen der Bewertung der zu testenden elliptischen Kurve sein kann. Weiterhin wird geprüft, ob die getwistete elliptische Kurve weder supersingulär noch anormal ist und ob die Punktezahl ord(E') dem Frey-Rück-Kriterium und dem Menezes-Okamoto-Vanstone-Kriterium genügt.

[0079] Ist für die getwistete elliptische Kurve gemäß dem Prüfungsschritt **105** ermittelt worden, dass das kryptographische Gütekriterium nicht erfüllt ist, so wird in den zuvor beschriebenen Schritt **104** verzweigt und es wird eine neue elliptische Kurve als zu testende elliptische Kurve aus der Menge gespeicherter elliptischer Kurven ausgewählt und das Verfahren wird in Schritt **102** wieder neu begonnen.

[0080] Die in den Schritten **103** bzw. **105** getestete elliptische Kurve E wird verworfen und in dem weiter beschriebenen kryptographischen Verfahren nicht eingesetzt.

[0081] Genügt jedoch auch die getwistete elliptische Kurve gemäß dem Prüfungsschritt **105** dem kryptographische Gütekriterium, so wird in einem weiterem Verfahrensschritt (Schritt **106**) die zu testende elliptische Kurve als im Weiteren verwendete elliptische Kurve ausgewählt und für ein in einem Weiteren Verfahrens-

schritt (Schritt **107**) durchgeführtes kryptographisches Verfahren bereitgestellt.

[0082] Im Weiteren wird im Rahmen eines kryptographischen Verfahrens (Schritt **107**) für die ermittelte elliptische Kurve beschrieben, wie ein skalares Vielfaches eines Punktes, der auf der ausgewählten elliptischen Kurve liegt, unter ausschließlicher Verwendung der x-Koordinate des Punktes hierzu berechnet wird.

[0083] Es ist in diesem Zusammenhang anzumerken, dass erfindungsgemäß die elliptische Kurve in jedem anderen kryptographischen Verfahren auch unter Verwendung der jeweiligen y-Koordinate eines auf der elliptischen Kurve sich befindlichen Punktes bei Transformationen oder bei Ermittlung eines skalaren Vielfaches des jeweiligen Punktes auf der elliptischen Kurve eingesetzt werden kann.

[0084] Es ist ferner anzumerken, dass die Reihenfolge, ob die elliptische Kurve E oder die zu der zu testenden elliptischen Kurve getwistete elliptische Kurve das kryptographische Gütekriterium erfüllt, unerheblich ist. Es kommt gemäß diesem Ausführungsbeispiel lediglich darauf an, dass sowohl die elliptische Kurve selbst als auch deren getwistete elliptische Kurve hinsichtlich der Erfüllung des kryptographischen Gütekriteriums untersucht werden.

[0085] Im Weiteren wird zur vereinfachten Darstellung des Ausführungsbeispiels jedoch lediglich das erfindungsgemäße Verfahren zur Multiplikation eines sich auf einer elliptischen Kurve befindenden Punktes unter ausschließlicher Verwendung der x-Koordinaten beschrieben.

[0086] Anders ausgedrückt wird gemäß dem im Weiteren beschriebenen Verfahren ein Vielfaches $n \cdot P$ eines Punktes P, der sich auf einer elliptischen Kurve befindet, berechnet.

[0087] Der Skalar $n = (n_1, ..., n_l)_2$, gegeben in Binärdarstellung, wird bitweise (beginnend beim sogenannten Most Significant Bit(MSB)n_1) abgearbeitet, d.h. bitweise mit dem Punkt P multipliziert.

[0088] Als Zwischenergebnisse werden in der jeweiligen i-ten Runde (i-ten Iteration) die Punkte $Q_i = m \cdot P$ und $R_i = (m + 1) \cdot P$ berechnet mit $n = (n_1, ..., n_\lambda)_2$ gemäß folgender Vorschriften, welche in einem Pseudocode dargestellt sind:

```
Q0 ← O;                    /* Initialisierung */
R0 ← P;
for i ← 1 to ℓ do /* Hauptschleife */
     if ni = 1 then
          Qi ← Qi−1 + Ri−1;
          Ri ← 2 · Ri−1;
     else
          Ri ← Ri−1 + Qi−1;
          Qi ← 2 · Qi−1;
     fi
od
return Qℓ; /* Ergebnis n · P */
```

[0089] Gemäß dem oben dargelegten ersten Teil-Verfahren wird zunächst einem Initialisierungspunkt Q_0 der Wert O zugeordnet, was einer Initialisierung dieser Variable entspricht.

[0090] Einer zusätzlichen Variable R wird in einem zusätzlichen Initialisierungsschritt als Initialisierungsvariable R_0 der Wert des Punktes P zugeordnet.

[0091] In einem zusätzlichen Schritt wird in der eigentlichen Berechnungsschleife in jeder Iteration für den jeweils berücksichtigten Skalarwert n_i für den Fall, dass der Skalarwert n_i den Wert 1 aufweist, dem Wert einer ersten Zwischenvariable Q zu der Iteration i (bezeichnet als Q_i) die Summe des Wertes der ersten Zwischenvariable Q_{i-1} zu der vorangegangen Iteration i-i und dem Wert der zweiten Zwischenvariable R_{i-1} zu der vorangegangen Iteration i-1 zugeordnet. Dem Wert der zweiten Zwischenvariable R_i in der Iteration i wird der zwei-

fache Wert der zweiten Zwischenvariable zu der vorangegangen Iteration i-1 zugeordnet.

[0092] Ist der Wert des Skalar n_i nicht gleich 1, so wird der zweiten Zwischenvariable R_i in der Iteration i die Summe der Werte der Summe der ersten Zwischenvariable R in vorangegangen Iteration i-1 und dem Wert ersten Zwischenvariable Q in der vorangegangen Iteration i-1 zugeordnet.

[0093] Der ersten Zwischenvariable Q_i zu der Iteration i wird der zweifache, d.h. der verdoppelte Wert der ersten Zwischenvariable Q zu der vorangegangenen Iteration i-1 zugeordnet.

[0094] Sind alle Skalarwerte n_i des Skalars n abgearbeitet, so wird der sich ergebende Wert der ersten Zwischenvariable Q_λ in der letzten Iteration λ, als Ergebniswert dieser Operation ausgegeben.

[0095] Wird das oben beschriebene Verfahren mit den üblichen Formeln zur Addition und Verdoppelung von Punkten auf einer elliptischen Kurve implementiert, so ist dieses üblicherweise langsamer als das ebenfalls bekannte und einsetzbare Doubleand-Add-Verfahren, weil unabhängig von dem jeweils aktuellen Bit des Skalars in jeder Runde, d.h. in jeder Iteration, sowohl addiert als auch verdoppelt werden muss.

[0096] Es ist aber ausreichend, wenn in jeder Runde nur die x-Koordinate der ersten Zwischenvariable Q_i und der zweiten Zwischenvariable R_i in einer jeweiligen Iteration i verwendet und neu berechnet werden.

[0097] Die dazu erforderlichen Additionsformeln werden im Weiteren hergeleitet und näher erläutert.

[0098] Auf diese Weise erhält man ein Verfahren zur Skalar-Multiplikation, welches erheblich schneller ist als die durchschnittliche Laufzeit des an sich bekannten Double-and-Add-Verfahrens. Zudem ist der Ablauf der Punkt-Multiplikation unabhängig von dem Wert des Skalars vollkommen gleichförmig. Es werden in der Fallunterscheidung lediglich die erste Zwischenvariable Q_i und die zweite Zwischenvariable R_i in Abhängigkeit von dem jeweils nächsten Bit des Skalars vertauscht. Daher ist der resultierende Algorithmus sicher gegen die Simple-Power-Analyse und die Timing-Angriffe.

[0099] Im Weiteren wird das Verfahren zum Berechnen der Skalar-Multiplikation in der Punktegruppe der ermittelten elliptischen Kurve über dem endlichen Körper für die üblicherweise verwendeten endlichen Körper GF(2m), d.h. einem Galois-Feld GF(2m) und Z/p·Z realisiert wird.

Punktdarstellung und Additionsformeln für endliche Körper der Charakteristik 2

[0100] Im Weiteren werden alle Punktdarstellungen und Additionsformeln für endliche Körper K der Charakteristik 2 beschrieben.

[0101] Anders ausgedrückt wird im Weiteren eine Transformationskurvengleichung einer elliptischen Kurve über einem Körper der Charakteristik k beschrieben.

[0102] Die Punkte nicht-supersingulärer Kurven über Körpern der Charakteristik 2 erfüllen die Gleichung:

$$y^2 + x \cdot y + x^3 + a_2 \cdot x^2 + a_6 = 0. \qquad (2)$$

[0103] Mit den in [8] beschriebenen affinen Additionsformeln ergeben sich folgende Gleichung für die X-Koordinaten der Summe $(x_1, y_1) + (x_2, y_2)$ zweier Punkte, einem ersten Punkt (x_1, y_1) und einem zweiten Punkt (x_2, y_2):

$$x_3 = \left(\frac{y_1 + y_2}{x_1 + x_2}\right)^2 + \frac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + a_2 \qquad (3)$$

[0104] Unter der Differenz $(x_1, y_1) - (x_2, y_2)$ der zwei Punkte:

$$x_4 = \left(\frac{y_1 + y_2 + x_2}{x_1 + x_2}\right)^2 + \frac{y_1 + y_2 + x_2}{x_1 + x_2} + x_1 + x_2 + a_2 \qquad (4)$$

[0105] Aus (3) erhält man:

$$x_3(x_1 + x_2)^2 = y_1^2 + y_2^2 + (y_1 + y_2)(x_1 + x_2) + \\ + (x_1 + x_2 + a_2)(x_1 + x_2)^2 \quad (5)$$

[0106] Analog folgt aus (4):

$$x_4(x_1 + x_2)^2 = y_1^2 + y_2^2 + x_2 + (y_1 + y_2 + x_2)(x_1 + x_2) + \\ + (x_1 + x_2 + a_2)(x_1 + x_2)2 \quad (6)$$

[0107] Aus (5) und (6) ergibt sich:

$$x_3 = \frac{x_1 \cdot x_2}{(x_1 + x_2)} + x_4 . \quad (7)$$

[0108] Aus der Additionsformel für die x-Koordinate des Doppelten eines Punktes P auf einer elliptischen Kurve E folgt:

$$x_5 = x_1^2 + \frac{a_6}{x_1^2} . \quad (8)$$

[0109] Die resultierenden Gleichungen (7) und (8) hängen nur noch von den x-Koordinaten der Punkte (x_1, y_1) und (x_2, y_2) ab. Damit kann das oben beschriebene Verfahren für Punkte in verschiedener Darstellung implementiert werden.

[0110] Um die relativ rechenaufwendige Inversion von Elementen des endlichen Körpers K einzusparen, wird erfindungsgemäß die sogenannte projektive Punktdarstellung verwendet.

[0111] In projektiver Schreibweise mit

$$x_i = \frac{X_i}{Z_i} \quad \text{für } 1 \leq i \leq 5$$

ergeben sich aus (6) die Gleichungen:

$$X_3 = X_4(X_1Z_2 + X_2Z_1)^2 + Z_4X_1Z_2X_2Z_1, \quad (9)$$

$$Z_3 = Z_4(X_1Z_2 + X_2Z_1)^2, \quad (10)$$

zur Addition zweier Punkte und aus Vorschrift (8) die Gleichungen

$$X_5 = \left(X_1^2 + \sqrt{a_6}Z_1^2\right)^2 , \quad (11)$$

$$Z_5 = X_1^2Z_1^2, \quad (12)$$

zur Verdoppelung eines Punktes P auf der elliptischen Kurve E, wobei $\sqrt{a_6}$ die (eindeutige) Quadratwurzel des Koeffizienten a_6 der elliptischen Kurvengleichung ist.

[0112] In dem oben beschriebenen Verfahren ist der Differenzpunkt ($R_i - Q_i$) zu einer Iteration i stets der Startpunkt P.

[0113] Anders ausgedrückt bedeutet dies, dass in den Formeln der Quotient $x_4 = X_4/Z_4$ immer die x-Koordinate des Startpunktes P auf der elliptischen Kurve ist.

[0114] Weil der Startpunkt P affin gegeben ist, gilt $Z_4 = 1$ und die Gleichungen (9), (10) vereinfachen sich weiter zu

$$X_3 = X_4(X_1Z_2 + X_2Z_1)^2 + X_1Z_2X_2Z_1, \quad (13)$$

$$Z_3 = (X_1Z_2 + X_2Z_1)^2. \quad (14)$$

[0115] Mit den oben dargestellten Additions- und Verdoppelungsformeln sind nach dem beschriebenen Ver-

fahren zur Berechnung des n-fachen eines Punktes noch $10(\lfloor \log n \rfloor + 1)$ Körpermultiplikationen und $3(\lfloor \log n \rfloor + 1)$ Körperadditionen notwendig.

[0116] Im Weiteren wird folgende Notation verwendet: der Quotient X_i/Z_i ist die x-Koordinate des i-fachen des Startpunktes P auf der elliptischen Kurve E. Insbesondere ist X_1 die x-Koordinate des Basispunktes in affiner Darstellung mit $Z_1 = 1$. Sei $x_4 = \frac{X_4}{Z_4}$ die eindeutig bestimmte Quadratwurzel des Koeffizienten a_6 der Kurvengleichung der elliptischen Kurve E.

[0117] Gemäß folgendem zweiten Teil-Verfahren wird die x-Koordinate des n-fachen des Basispunktes in projektiver Darstellung berechnet. Sei $n = (n_1, ..., n_\lambda)_2$ in Binärdarstellung mit n_1 als Most Significant Bit gegeben:

```
A ← 1; /* A = X_0 * //* Initialisierung * /
B ← 0; /* B = Z_0 * /
C ← X1; /* C = X_1 * /
```

```
D ← 1; /* D = Z_1 */
for i ← 1 to ℓ do /* Hauptschleife */
    if n_i = 1 then
        A ← A * D; /* A = X_n * Z_n + 1 */
        B ← B * C; /* B = X_n + 1 * Z_n */
        C ← C * C; /* C = (X_n + 1)^2 */
        D ← D * D; /* D = (Z_n + 1)^2 */
        E ← A * B; /* E = X_n * Z_n + 1 * X_n + 1 * Z_n */
        B ← A + B; /* B = X_n * Z_n + 1 + X_n + 1 * Z_n */
        B ← B * B; /* B = Z_2n + 1 */
        A ← X_1 * B;
        /* A = X_1 * (X_n * Z_n + 1 + X_n + 1 * Z_n)^2 */
        A ← A + E; /* A = X_2n + 1 */
        E ← √a_6 * D; /* E = (a_6)^0,5 * (Z_n + 1)^2 */
        D ← C * D; /* D = Z_2n + 2 */
        C ← C + E;
        /* C = (X_n + 1)^2 + (a_6)^0,5 * (Z_n + 1)^2 */
        C ← C * C; /* C = X_2n + 2 */
    else
        C ← B * C; /* C = X_n + 1 * Z_n */
        D ← A * D; /* D = X_n * Z_n + 1 */
        A ← A * A; /* A = (X_n)^2 */
        B ← B * B; /* B = (Z_n)^2 */
        E ← C * D; /* E = X_n + 1 * Z_n * X_n * Z_n + 1 */
        D ← C + D; /* D = X_n + 1 * Z_n + X_n * Z_n + 1 */
        D ← D * D; /* D = Z_2n + 1 */
        C ← X_1 * D;
        /* C = X_1 * (X_n + 1 * Z_n + X_n * Z_n + 1)^2 */
        C ← C + E; /* C = X_2n + 1 */
        E ← √a_6 * B; /* E = (a_6)^0,5 * (Z_n)^2 */
        B ← A * B; /* B = Z_2n */
        A ← A + E; /* A = (X_n)^2 + (a_6)^0.5 * (Z_n)^2 */
        A ← A * A; /* A = X_2n */
    fi
od
return (A, B); /* projektives Ergebnis */
```

[0118] Der resultierende Quotient A/B ist die x-Koordinate des Punktes n·P. Zum Schluss wird noch das multiplikative Inverse 1/B von B berechnet, um die x-Koordinate von n·P in affiner Darstellung bestimmen zu können. Zum Bestimmen des Inversen von 1/B wird der Fermatsche Satz verwendet. Details zum Bestimmen des Inversen von 1/B werden im Weiteren dargestellt.

[0119] Ein erheblicher Vorteil des zweiten Teil-Verfahrens besteht darin, dass keine Fallunterscheidungen zur Addition oder Verdoppelung des unendlich fernen Punktes O oder des Punktes der Ordnung 2 erforderlich sind.

[0120] Die Additionsformeln (13), (14) und (11), (12) bleiben auch für alle Sonderfälle dieser Art gültig.

[0121] Der Punkt P ist gleich (X_1, Z_1) der Basispunkt. In der obigen Punktdarstellung hat der unendlich ferne Punkt 0 die Darstellung O = (X, Z) mit $X \neq 0$ und $Z = 0$.

[0122] Im Weiteren werden einige mögliche Sonderfälle dargestellt:
– Sonderfall Addition von $(X_n, Z_n) = (X_m, Z_r)$: d.h. $(n + 1)\cdot P = n\cdot P$. wodurch folgt: P = O.
Dieser Fall kann nicht auftreten, weil in jeder Runde, d.h. in jeder Iteration gilt: $Q_i + P = R_i$.
– Sonderfall Addition von $(X_n, Z_n) = O$, $(X_m, Z_m) \neq O$: aus (13), (14) ergibt sich:

$$(X_{n+m}, Z_{n+m}) = X_1X_m^2Z_n^2, X_m^2Z_n^2) = x(P) \tag{15}$$

als das richtige Resultat.
– Sonderfall Addition von $(X_n, Z_n) = =$, $(X_m, Z_m) \neq O$: aus (13), (14) ergibt sich:

$$(X_{n+m}, Z_{n+m}) = (X_1X_n^2Z_m^2, X_n^2Z_m^2) = x(-P) \tag{16}$$

als das richtige Resultat.
– Sonderfall Addition von $(X_n, Z_n) = -(X_m, Z_m)$: da $X_n/Z_n = X_m/Z_m$ gilt und die Multiplikation nullteilerfrei ist, folgt $X_{n+m} \neq 0$, $Z_{n+m} = 0$ und $(n + m)\cdot P = O$ als das richtige Resultat.
– Sonderfall Addition von $(X_n, Z_n) \neq O$, $(X_m, Z_m) \neq O$ mit $2(X_n, Z_n) = 2(X_m, Z_m) = O$: Das heißt $2nP = 2(n + 1)P = O$, womit $2\cdot P = O$ folgt. Daher gilt entweder $(X_n, Z_n) = O$ oder $(X_m, Z_m) = O$, und der Fall kann nicht auftreten.
– Sonderfall Verdoppelung von $(X_n, Z_n) = O$: aus (11), (12) und der Nullteilerfreiheit der Multiplikation folgt $X_{2n} \neq 0$ und $Z_{2n} = 0$ und damit $(X_{2n}, Z_{2n}) = O$.
– Sonderfall Verdoppelung von (X_n, Z_n) mit $2(X_n, Z_n) = O$, d.h. $X_n = 0$ und $Z_n \neq 0$: aus (11), (12) und der Nullteilerfreiheit der Multiplikation folgt $X_{2n} \neq 0$ und $Z_{2n} = 0$ und daher $(X_{2n}, Z_{2n}) = 0$.

[0123] Damit kann dieses zweite Teil-Verfahren zur Skalar-Multiplikation auch dann angewendet werden, wenn der Skalar n größer als die Ordnung des Basispunktes P ist.

[0124] In der oben beschriebenen Form werden gemäß dem zweiten Teil-Verfahren nur die x-Koordinaten der Punkte $n\cdot P$ und $(n + 1)\cdot P$ berechnet. Falls der vollständige Ergebnispunkt einschließlich der y-Koordinaten berechnet werden soll, so ist es möglich, aus den berechneten x-Koordinaten und der Gleichung der elliptischen Kurve die y-Koordinate von $n\cdot P$ zu berechnen.

[0125] Seien im Folgendem x_1 die x-Koordinate des Ergebnispunktes $n\cdot P$, x_2 die x-Koordinate des Ergebnispunktes $(n + 1)\cdot P$, (x, y) die Koordinaten des Startpunktes P und y_1 die gesuchte y-Koordinate des Ergebnispunktes $n\cdot P$ (alle affin gegeben).

[0126] Dann folgt gemäß Vorschrift (3):

$$x_2 = \left(\frac{y_1 + y}{x_1 + x}\right)^2 + \frac{y_1 + y}{x_1 + x} + x_1 + x + a_2. \tag{17}$$

Wird der Term y^2 gemäß der Kurvengleichung (2) ersetzt und anschließend nach y_1 aufgelöst, so ergibt sich:

$$y_1 = y + (x_1 + x)\frac{(x_1 + x)(x_2 + x) + x^2 + y}{x} \tag{18}$$

für die gesuchte y-Koordinate, falls gilt: $x \neq 0$.

[0127] Für den Fall x = 0 ist das Ergebnis $n\cdot P$ entweder der unendlich weit entfernte Punkt O oder der (eindeutig bestimmte) Punkt $(0, \sqrt{a_6})$der Ordnung 2.

[0128] Für eine Implementierung mit projektiver Koordinatendarstellung ergibt sich (im Fall $x \neq 0$) gemäß der oben gewählten Notation mit $x_1 = X_1/Z_1$ und $x_2 = X_2/Z_2$ die Vorschrift:

$$y_1 = \frac{(x_1 + xz_1)\left((x_1 + xz_1)(x_2 + xz_2) + z_1z_2\left(x^2 + y\right)\right) + z_1^2z_2xy}{z_1^2z_2x} \quad (19)$$

für die y-Koordinate des Punktes n·P in projektiver Koordinatendarstellung.

Punktdarstellung und Additionsformeln für endliche Körper mit Charakteristik größer als 3.

[0129] Im Weiteren werden Punktdarstellung und Additionsformeln für endliche Körper mit Charakteristik größer als 3 erläutert.

[0130] Die Punkte nicht-supersingulärer Kurven über endlichen Körpern der Charakteristik größer als 3 erfüllen die Gleichung

$$y^2 = x^3 + a_4 \cdot x + a_6. \quad (20)$$

[0131] Aus den Additionsformeln für Punkte auf einer elliptischen Kurve, wie sie beispielsweise in [8] beschrieben sind, ergeben sich folgende Gleichungen für die x-Koordinaten der Summe $(x_1, y_1) + (x_2, y_2)$ zweier Punkte, einem ersten Punkt (x_1, y_1) und einem zweiten Punkt (x_2, y_2) auf einer elliptischen Kurve

$$x_3 = \left(\frac{y_1 - y_2}{x_1 - x_2}\right)^2 - x_1 - x_2, \quad (21)$$

und der Differenz $(x_1, y_1) - (x_2, y_2)$

$$x_4 = \left(\frac{y_1 + y_2}{x_1 - x_2}\right)^2 - x_1 - x_2. \quad (22)$$

[0132] Aus der Summe der Vorschriften (21) und (22) erhält man durch Substitution der Terme y_1^2 und y_2^2 mittels Vorschrift (20) die folgende Vorschrift:

$$(x_3 + x_4)(x_1 - x_2)^2 = 2(x_1 + x_2)(x_1x_2 + a_4) + 4a_6 \quad (23)$$

für die Punktaddition.

[0133] Analog folgt aus der Additionsformel für die x-Koordinate des Doppelten eines Punktes (x_1, y_1) durch Substitution des Terms y_1^2 mittels Vorschrift (20) die folgende Vorschrift:

$$4x_5(x_1^3 + a_4x_1 + a_6) = (x_1^2 - a_4)^2 - 8a_6x_1. \quad (24)$$

[0134] In projektiver Schreibweise mit $x_i = X_i/Z_i$ für $1 \le i \le 5$ und $Z_4 = 1$ erhält man aus Vorschrift (23) die folgenden Vorschriften:

$$X_3 = 2(X_1Z_2 + X_2Z_1)(X_1X_2 + a_4Z_1Z_2) + 4a_6Z_1^2Z_2^2 - X_4(X_1Z_2 - X_2Z_1)^2, \quad (25)$$

$$Z_3 = (X_1Z_2 - X_2Z_1)^2, \quad (26)$$

zur Addition zweier Punkte und aus Vorschrift (24) die Vorschriften

$$X_5 = (X_1^2 - a_4Z_1^2 - 8a_6X_1Z_1^3, \quad (27)$$

$$Z_5 = 4X_1Z_1(X_1^2 + a_4Z_1^2) + 4a_6Z_1^4, \quad (28)$$

zur Verdoppelung eines Punktes auf der elliptischen Kurve.

[0135] Die Vorschriften (25), (26) können mit 10 Multiplikationen und 7 Additionen und Subtraktionen ausgewertet werden. Die Vorschriften (27), (28) können mit 9 Multiplikationen und 8 Additionen und Subtraktionen ausgewertet werden.

[0136] Mit diesen Additionsvorschriften und Verdoppelungsvorschriften, wie sie oben beschrieben wurden, sind gemäß dem in [7] beschriebenen Verfahren zur Berechnung des n-fachen eines Punktes $19(\lfloor \log n \rfloor + 1)$ Körpermultiplikationen und $15(\lfloor \log n \rfloor + 1)$ Körperadditionen und Körpersubtraktionen erforderlich (zuzüglich einmalig 15 Multiplikationen, 7 Additionen und Subtraktionen und einer Inversion zur Berechnung des Ergebnispunktes in affiner Koordinatendarstellung)

[0137] Im Weiteren wird das Teil-Verfahren zur Berechnung des skalaren Vielfachen eines affinen gegebenen Punktes P = (x, y) beschrieben.

[0138] Die x-Koordinate des Punktes P bildet dabei den Wert X_4 in den Vorschriften (25) und (26). Es ist in diesem Zusammenhang anzumerken, dass auch die Vereinfachung $Z_4 = 1$ möglich ist, weil der Punkt P stets affin gegeben ist. Damit sind X_1/Z_1 und X_2/Z_2 die x-Koordinaten von zwei Punkten, die sich um den Punkt P unterscheiden.

[0139] Im Folgenden wird mit (X_m, Z_m) die projektiv gegebene x-Koordinate des Punktes $m \cdot P$ bezeichnet. Sei X_1 die x-Koordinate des Basispunktes P in affiner Darstellung (d.h. $Z_1 = 1$), und sei $n = (n_1, ..., n_\lambda)_2$ die Binärdarstellung des Skalars n mit n_1 als Most Significant Bit.

[0140] Gemäß dem im Weiteren in einem Pseudo-Code dargestellten dritten Teil-Verfahren wird die x-Koordinate des n-fachen des Basispunktes P in projektiver Darstellung berechnet:

```
procedure double (var Ã, B̃, C̃, D̃)

/* Input : C̃ / D̃ = X_n / Z_n, Output : Ã / B̃ = X_2n / Z_2n */
```

```
G ← D̃ * D̃;      /* G = Z_n^2 */
H ← a4 * G;      /* H = a4 * Z_n^2 */
B̃ ← C̃ * D̃;      /* B̃ = X_n * Z_n */
B̃ ← B̃ + B̃;      /* B̃ = 2 * X_n * Z_n */
Ã ← C̃ * C̃;      /* Ã = X_n^2 */
I ← Ã + H;       /* I = X_n^2 + a4 * Z_n^2 */
256Z_n^16(4a4^3 + 27a6^2)^2 (X_n^2 + a4 * Z_n^2)* /
Ã ← Ã - H;       /* Ã = X_n^2 - a4 * Z_n^2 */
Ã ← Ã * Ã;       /* Ã = (X_n^2 - a4 * Z_n^2)^2 */
H ← a6 * G,      /* H = a6 * Z_n^2 */
H ← H + H;       /* H = 2 * a6 * Z_n^2 */
B̃ ← B̃ * H;       /* B̃ = 4 * a6 * X_n * Z_n^3 */
Ã ← Ã - B̃;
/* Ã = (X_n^2 - a4 * Z_n^2)^2 - 4 * a6 * X_n * Z_n^3 */
Ã ← Ã - B̃;
/* X_2n = (X_n^2 - a4 * Z_n^2)^2 - 8 * a6 * X_n * Z_n^3 */
H ← G * H;       /* H = 2 * a6 * Z_n^4 */
B̃ ← H + I;
/* B̃ = 2 * Z_n * (X_n * (X_n^2 + a4 * Z_n^2) + a6 * Z_n^3)* /
B̃ ← B̃ + B̃;
/* Z_2n = 4 * Z_n * (X_n * (X_n^2 + a4 * Z_n^2) */
/*+ a6 * Z_n^3)* /
return;


procedure add (var Ã, B̃, C̃, D̃, Ẽ, F̃)
/* Input : C̃ / D̃ = X_n / Z_n, Ẽ / F̃ = X_m / Z_m, * /
```

```
/*Output : Ã / B̃ = X_n + m / Z_n + m */
if D̃ = 0 then
     Ã ← Ẽ;     /* X_n / Z_n ist Punkt O */
     B̃ ← F̃;
     return;
fi;
if F̃ = 0 then
     Ã ← C̃;     /* X_m / Z_m ist Punkt O */
     B̃ ← D̃;
return;
fi;
G ← C̃ * Ẽ;      /* G = X_n * X_m */
H ← D̃ * F̃;      /* H = Z_n * Z_m */
I ← a4 * H;      /* I = a4 * Z_n * Z_m */
G ← G + I;       /* G = X_n * X_m + a4 * Z_n * Z_m */
I ← C̃ * F̃;      /* I = X_n * Z_m */
Ã ← D̃ * Ẽ;      /* Ã = X_m * Z_n */
B̃ ← I - Ã;      /* B̃ = X_n * Z_m - X_m * Z_n */
B̃ ← B̃ * B̃;      /* Z_n + m = (X_n * Z_m - X_m * Z_n)^2 */
I ← Ã + I;      /* I = X_n * Z_m + X_m * Z_n */
G ← G * I;      /* G = (X_n * Z_m + X_m * Z_n) * */
/*  * (X_n * X_m + a4 * Z_n * Z_m) */
H ← H * H;      /* H = Z_n^2 * Z_m^2 */
H ← a6 * H;      /* H = a6 * Z_n^2*Z_m^2 */
H ← H + H;      /* H = 2 * a6 * Z_n^2 * Z_m^2 */
G ← G + H;      /* G = (X_n * Z_m + X_m * Z_n) * */
/*  * (X_n * X_m + a4 * Z_n * Z_m) + 2 * a6 * Z_n^2 * Z_m^2 */
G ← G + G;      /* G = 2 * (X_n * Z_m + X_m * Z_n) */
```

```
/*
* (X _ n * X _ m + a4 * Z _ n * Z _ m) + 4 * a6 * Z _ n^2 * Z_m^2 * /EIN
BETTEN
H ← X1 * B̃;     /* H = X_1*(X_n*Z_m-X_m*Z_n)^2 */
Ã ← G − H;      /* X_n+m = 2*(X-n*Z_m+X_m*Z_n)* */
                /* *(X_n*X_m+a4*Z_n*Z_m)+4*a6*Z_n^2*Z_m^2- */
                /* -X_1*(X_n*Z_m-X_m*Z_n)^2 */
return;

/* Input: X-1, Output: A/B=X_n/Z_n, C/D=X_n+1/Z_n+1 */
A ← 1;          /* A = X_0=, Punkt O */
B ← 0;          /* B = Z_0 */
C ← X1;         /* C = X_1, Basispunkt */
D ← 1;          /* D = Z_1 */
for i ← 1 to λ do
     if ni = 1 then
          add(A, B, A, B, C, D);
          double(C, D, C, D);
     else
          add(C, D, C, D, A, B);
          double(A, B, A, B);
     fi;
od;
```

[0141] Gilt B = 0, so ist n·P = O der unendlich ferne Punkt, wobei anzumerken ist, dass der unendlich ferne Punkt O die Darstellung (X, Z) hat mit X ≠ 0 und Z = 0. Ist B ≠ 0, so ist A/B mod p die x-Koordinate des Ergebnispunktes n·P. Soll auch die y-Koordinate von n·P bestimmt werden, so wird folgender Zusammenhang benutzt: Seien (x_1, y_1) der Basispunkt P, x_n = A/B die x-Koordinate von n·P und x_{n+1} = C/D die x-Koordinate von (n + 1)·P. Aus der Additions-Vorschrift (21) ergibt sich mittels Substitution der Terme y_1^2 und y_n^2 durch die Kurvengleichung der elliptischen Kurve (20):

$$y_n = \frac{2a_6 + (x_1x_n + a_4)(x_1 + x_n) - x_{n+1}(x_1 - x_n)^2}{2y_1}. \qquad (29)$$

[0142] In projektiver Schreibweise erhält man für die Ergebnisse des dritten Teil-Verfahrens die Gleichung

$$y_n = \frac{2a_6z_n^2z_{n+1} + z_{n+1}(x_1x_n + a_4z_n)(x_n + x_1z_n) - x_{n+1}(x_n - x_1z_n)^2}{2y_1z_n^2z_{n+1}} \qquad (30)$$

Gemäß dem im Weiteren beschriebenen und in einem Pseudo-Code erläuterten vierten Teil-Verfahren wird der Punkt n·P = (x_1, y_1) in affiner Darstellung aus den bereits berechneten Werten A, B, C und D berechnet:

```
/* Input A, B, C, D, Output: (x1, y1) = (A, B) oder O */
G ← X1 * B;     /* G = X_1*Z_n */
H ← X1 * A;     /* H = X_1*X_n */
I ← a4 * B;     /* I = a4*Z_n */
H ← H + I;      /* H = X_1*X_n+a4*Z_n */
I ← A + G;      /* I = X_n+X_1*Z_n */
H ← H * I;      /* H = (X_1*X_n+a4*Z_n)(X_n+X_1*Z_n) */
G ← A - G;      /* G = X_n-X_1*Z_n */
I ← G * G;      /* I = (X_n-X_1*Z_n)^2 */
I ← C * I;      /* I = X_n+1*(X_n-X_1*Z_n)^2 */
G ← B * B;      /* G = Z_n^2 */
G ← a6 * G;     /* G = a6*Z_n^2 */
G ← G + G;      /* G = 2*a6*Z_n^2 */
G ← G + H;      /* G=2*a6*Z_n^2+(X_1*X_n+a4*Z_n)(X_n+X_1*Z_n)*/
G ← D * G;      /* G = Z_n+1*(2*a6*Z_n^2+(X_1*X_n+a4*Z_n)* */
                /* *(X_n+X_1*Z_n)) */
C ← G - I;      /* C = Z_n+1*(2*a6*Z_n^2+(X_1*X_n+a4*Z_n)* */
                /* *(X_n+X_1*Z_n))-X_n+1*(X_n-X_1*Z_n)^2 */
G ← B * D;      /* G = Z_n*Z_n+1 */
G ← Y1 * G;     /* G = Y_1*Z_n*Z_n+1 */
G ← G + G;      /* G = 2*Y_1*Z_n*Z_n+1 */
H ← B * G;      /* H = 2*Y_1*Z_n^2*Z_n+1 */
if B = 0 then
     Ergebnis ist O;
else if D = 0 then
     A ← X1;   /* Ergebnis ist -P = (X1,-Y1) */
     B ← -Y1;
else
     H ← modinv(H);      /* H = 1/(2*Y_1*Z_n^2*Z_n+1) */
     B ← C * H;      /* B = y_n */
     H ← G * H;      /* H = 1/Z_n */
     A ← A * H;      /* H = x_n */
fi;
```

[0143] Das affine Ergebnis $n \cdot P = (x_n, y_n)$ steht in den im Weiteren beschriebenen Registern (A, B), welche die Werte A und B speichern. Insbesondere ist insgesamt nur eine modulare Inversion zur Berechnung der affinen x-Koordinate und y-Koordinate des Ergebnispunktes erforderlich.

[0144] Solange der unendlich ferne Punkt O nicht als Zwischenergebnis der Hauptschleife in dem oben beschriebenen vierten Teil-Verfahren auftritt (d.h. solange gilt $B \neq 0$ und $D \neq 0$), solange ist der Ablauf des vierten Teil-Verfahrens vollkommen gleichförmig und von konkreten Werten des Skalars n unabhängig.

[0145] Bei der Bearbeitung der einzelnen Bits n_i des Skalars n werden in jedem Durchlauf (d.h. in jeder Iteration) die gleiche Anzahl von Körperoperationen in der gleichen Reihenfolge durchgeführt. Es werden lediglich in Abhängigkeit von dem Wert des aktuellen Bits n_i des Skalars n Zugriffe auf das Registerpaar (A, B), welches die Werte A bzw. B gespeichert hat, mit Zugriffen auf das Registerpaar (C, D), welches die zuvor ermittelten Werte C und D gespeichert hat, vertauscht. Das vierte Teil-Verfahren ist daher vor Timing-Angriffen geschützt.

[0146] Falls nicht ausgeschlossen werden kann, dass während der Berechnung der Fall B = 0 oder D = 0 eintritt, so sollten in die entsprechenden Fallunterscheidungen der oben beschriebenen Routine-„Add"-Dummy-Befehle vorgesehen werden, um den gleichförmigen Ablauf der Skalar-Multiplikation nicht zu gefährden.

[0147] Das gleiche gilt auch für die Fallunterscheidungen bei der abschließenden Berechnung der affinen Darstellung des Ergebnispunktes, d.h. für die Fälle, bei denen das Ergebnis der unendlich ferne Punkt O oder der Wert „-P" ist.

[0148] Die oben erläuterten Additions-Vorschriften (25), (26) und (27), (28) für die Skalar-Multiplikation bleiben für alle Sonderfälle gültig, bis auf den Fall der Addition des unendlich fernen Punktes O.

[0149] Sei $P = (X_1, Z_1)$ der Basispunkt:
– Addition von $(X_n, Z_n) = (X_m, Z_m)$: Das heißt $(n + 1)\cdot P = n\cdot P$, womit P = O folgt. Dieser Fall kann nicht auftreten.
– Addition von $(X_m, Z_m) = -(X_m, Z_m) \neq O$: Da $X_n/Z_n = X_m/Z_m$ gilt, folgt $Z_{n+m} = 0$. Damit auch $X_{n+m} = 0$ gilt, muss (X_n, Z_n) gemäß Vorschrift (27), (28) die x-Koordinate eines Punktes der Ordnung 2 sein. Dann folgt aber aus den Voraussetzungen $n\cdot P + P = -n\cdot P$. dass $Z_n = 0$ ist. Dieser Fall kann also ebenfalls nicht auftreten und es gilt stets $X_{n+m} \neq 0$.
– Addition von $(X_n, Z_n) \neq 0$, $(X_m, Z_m) \neq O$ mit $2(X_n, Z_n) = 2(X_m, Z_m) = O$: Das heißt $2nP = 2(n + 1)P = O$, womit 2P = O folgt. Daher gilt entweder $(X_n, Z_n) = O$ oder $(X_m, Z_m) = O$ und der Fall kann ebenfalls nicht auftreten.
– Verdoppelung von $(X_n, Z_n) = O$: aus den Vorschriften (27), (28) und der Nullteilerfreiheit der Multiplikation folgt $X_{2n} \neq 0$ und $Z_{2n} = 0$ und damit $(X_{2n}, Z_{2n}) = O$.
– Verdoppelung von (X_n, Z_n) mit $2(X_n, Z_n) = O$: Aus den Vorschriften (27), (28) folgt direkt, dass $Z_{2n} = 0$ gilt. Wird das Ergebnis der Resultante aus den Gleichungen für X_{2n}, Z_{2n} berechnet, so ergibt sich $256Z_n^{16}(4_4^3 + 27a_6^2)^2$. Weil die elliptische Kurve keine singulären Punkte aufweist, folgt daraus $(X_{2n}, Z_{2n}) = O$.

Punktdarstellung und Additionsformeln für endliche Körper der Charakteristik 3

[0150] Im Weiteren werden eine Punktdarstellung und Additionsformeln für endliche Körper K der Charakteristik 3 beschrieben.

[0151] Analog zu der in den obigen Abschnitten hinsichtlich der Punktdarstellung und Additionsformeln für endliche Körper der Charakteristik 2 und größer als 3 beschriebenen Vorgehensweisen können entsprechende Vorgehensweisen gemäß diesem Ausführungsbeispiel auch für den Fall eines endlichen Körpers der Charakteristik 3 mit entsprechenden Formeln zur Addition und Verdoppelung von Punkten für elliptische Kurven über einem endlichen Körper verwendet werden, welche im Folgenden hergeleitet und näher erläutert Werden.

[0152] Die Punkte nicht-supersingulärer Kurven über endlichen Körpern der Charakteristik 3 erfüllen die Gleichung

$$y^2 = x^3 + a_2\cdot x^2 + a_6. \tag{31}$$

[0153] Aus den oben beschriebenen affinen Additionsformeln ergeben sich mit einer analogen Herleitungsweise wie oben beschrieben und unter Verwendung analoger Notation die Formeln

$$(x_3 + x_4)(x_1 - x_2)^2 = a_6 + a_2x_1x_2 - x_1x_2(x_1 + x_2), \tag{32}$$

zur Addition und

$$x_5 = \frac{x_1^4 + a_6 \cdot x_1 - a_2 \cdot a_6}{x_1^3 + a_2 \cdot x_1^2 + a_6}, \tag{33}$$

für die x-Koordinate des Doppelten eines Punktes auf der elliptischen Kurve.

[0154] Die algorithmische Umsetzung der Vorschriften (32) und (33) in einem Verfahren zur Skalar-Multiplikation erfolgt vollkommen analog zu den oben beschriebenen Teil-Verfahren.

[0155] Wie in [6] beschrieben ist, ermöglicht die Punktdarstellung mit projektiven Koordinaten eine einfache Methode zum Schutz gegen die Differential-Power-Analyse. Hierzu wird während der Initialisierung der oben beschriebenen Teil-Verfahren die Variable A nicht auf den Wert 1, sondern auf einen zufälligen, von dem Wert „0" verschiedenen Wert gesetzt, die Variable D auf einen weiteren zufälligen, von „0" verschiedenen Wert gesetzt und die Variable C mit dem Wert von D multipliziert.

[0156] Es ergibt sich somit:
A ← RAND; /* A = X_0 */ /* Initialisierung */
B ← 0; /* B = Z_0 */
D ← RAND; /* D = Z_1 */
C ← D * X_1; /* C = X_1 * Z_1 */

[0157] Die Funktion RAND wird gemäß diesem Ausführungsbeispiel folgendermaßen implementiert:
Zunächst wird mittels eines Zufallszahlen-Generators eine Zufallszahl generiert und anschließend wird das Least Significant Bit auf den Wert „1" gesetzt, um zu gewährleisten, dass der Wert von dem Wert „0" verschieden ist.

[0158] Mit dieser Maßnahme werden die projektiven Darstellungen der Startwerte und aller Zwischenergebnisse mit für jede Anwendung der Skalar-Multiplikation neugewählten zufälligen Werten erweitert. Dadurch werden Korrelationen zwischen Eingabewerten, dem geheimen Skalar und den berechneten Zwischenergebnissen, die ein Angreifer für eine statistische Analyse des Stromverbrauchprofils verwenden könnte, verhindert.

[0159] Differential-Power-Analyse, wie sie in [5] beschrieben ist, ist in diesem Fall nicht mehr möglich.

[0160] Gerade für kontaktlose Chip-Karten sind Angriffe, die fehlerhafte Ergebnisse von kryptographischen Algorithmen ausnutzen, eine besondere Bedrohung, da die relativ instabile Energieversorgung von kontaktlosen Karten leicht zu Fehlfunktionen aller Art führen kann.

[0161] Um solche Differential-Fault-Angriffe abwehren zu können, werden üblicherweise redundante Relationen auf den Berechnungsergebnissen eingeführt, die am Ende der Berechnung einen einfachen Korrektheitstest der Ergebnisse ermöglichen. Bei elliptischen Kurven ist beispielsweise der Test, ob der Ergebnispunkt einer Skalar-Multiplikation immer noch die Kurvengleichung der elliptischen Kurve erfüllt, eine solche Gegenmaßnahme.

[0162] Bei der erfindungsgemäß gewählten Punktdarstellung ist es aber aufwendig, die y-Koordinate zu bestimmen und damit diesen Test durchzuführen.

[0163] Stattdessen wird erfindungsgemäß eine andere Abwehrmaßnahme gegen einen Differential-Fault-Angriff vorgesehen: Bei allen in den oben beschriebenen Teil-Verfahren erfindungsgemäß verwendeten Additionsformeln werden die Parameter der Kurvengleichung der ermittelten elliptischen Kurve in einer Art und Weise verwendet, die mögliche Gruppenstrukturen von fehlerhaften Punkten und Kurvengleichungen weitestgehend festlegt.

[0164] Ein fehlerhaftes Ergebnis kann nur auf einer elliptischen Kurve liegen, deren Gruppenstruktur entweder isomorph zur gewünschten elliptischen Kurve oder zu der zugehörigen getwisteten elliptischen Kurve ist.

[0165] Erfindungsgemäß werden daher nur elliptische Kurven verwendet, bei denen sowohl die Gruppenstruktur der gegebenen elliptischen Kurve, d.h. der ermittelten elliptischen Kurve, als auch die der ermittelten elliptischen Kurve zugehörigen getwisteten Kurve kryptographisch stark ist. Dadurch kann vollständig auf Plausibilitätstests für Berechnungsergebnisse verzichtet werden.

[0166] Ein Angreifer kann durch fehlerhafte Ergebnisse prinzipiell nur wenige Bits des geheimen Skalars ermitteln. Es ist in diesem Zusammenhang anzumerken, dass bei optimal gewählten Kurvenparametern für Körper der Charakteristik 2 es genau 2 Bits sind und für Körper der Charakteristik größer als 3 kann ein Angreifer bei optimaler Wahl der Parameter mit diesen Angriffen keine Bits des geheimen Skalars erfahren.

[0167] Nach Durchführung der oben beschriebenen Teil-Verfahren für die Skalar-Multiplikation wird erfindungsgemäß noch die projektiv gegebene x-Koordinate, welche in den Registern A und B gespeichert ist, des Ergebnispunktes in eine affine Darstellung gebracht.

[0168] Dazu ist eine Division bzw. eine Inversion notwendig. Erfindungsgemäß wird im Falle eines Körpers $GF(2^m)$ der Charakteristik 2 folgender Algorithmus zur Inversion eingesetzt, wobei der kleine Fermatsche Satz verwendet wird.

[0169] Der Wert, welcher in dem Register B gespeichert ist, soll invertiert werden. Dabei dient Register C zur Speicherung der Zwischenergebnisse.

[0170] Das Ergebnis der Inversion ist am Ende des Teil-Verfahrens in dem Register C gespeichert.

```
C ← B * B;
for i ← 1 to m-2 do
     C ← B * C;
     C ← C * C;
od
```

[0171] Nach der Durchführung beider Algorithmen, d.h. der Algorithmen der Teil-Verfahren zur Skalar-Multiplikation und anschließend des Teil-Verfahrens zur Inversion des Wertes von B, kann die affine x-Koordinate des Ergebnispunktes mit dem folgenden Befehl berechnet werden:

$A \leftarrow A*C$.

[0172] Die affine x-Koordinate des Ergebnispunktes ist am Ende des Teil-Verfahrens in dem Register A gespeichert.

[0173] Wenn der vollständige affine Ergebnispunkt der Skalar-Multiplikation, d.h. sowohl die x-Koordinate als auch die y-Koordinate des Ergebnispunktes berechnet werden sollen, so ist dafür erfindungsgemäß nur eine Ausführung des Inversions-Algorithmus erforderlich.

[0174] Wie bereits oben aufgeführt wurde, sind mehrere Fälle zu unterscheiden:

1. Gilt nach der Skalar-Multiplikation Register B = 0, dann ist der Ergebnispunkt der unendlich ferne Punkt O.
2. Gilt nach erfolgter Skalar-Multiplikation Register D = 0, dann ist der Ergebnispunkt der inverse Startpunkt. Das heißt, falls der Startpunkt die Koordinaten (x, y) hat, dann hat der Ergebnispunkt die Koordinaten (x, y + x).
3. In allen anderen Fällen gilt $x \neq 0$, $B \neq 0$ und $D \neq 0$. Es wird mittels des oben beschriebenen Verfahrens zur Inversion des Körperelements berechnet: $\alpha = (xZ_1^2Z_2)(= (xB^2D)^{-1}$ in der Registernotation). Dann gilt $Z_1^{-1} = \alpha Z_1 Z_2$ und die Koordinaten des Ergebnispunktes können mit Vorschrift (19) für die y-Koordinate in affiner Darstellung berechnet. Insbesondere ist erfindungsgemäß nur eine Inversion notwendig, um beide Koordinaten zu berechnen.

[0175] Für Körper der Charakteristik 3 und größer als 3 können der oben beschriebene Algorithmus zur Inversion von Körperelementen und die Fallunterscheidungen zur Berechnung der vollständigen Ergebnispunktes entsprechend angepasst werden.

[0176] Im folgenden wird ein Beispiel für eine geeignete elliptische Kurve gegeben, welche auf die oben beschriebene Weise ermittelt worden ist.

[0177] Die Parameter
a4 = 160759863572385372610176464653486248077700068594 9
a_6 = 266467061243246186834812072239672494440669583190 7
definieren eine elliptische Kurve über dem endlichen Körper
Z/5846006848301083051854307689248681887734956450399

[0178] Die von der elliptischen Kurve erzeugte Punktegruppe hat die Ordnung
5846006848301083051854312524952084000434031723889,

die von der getwisteten elliptischen Kurve erzeugte Punktegruppe hat die Ordnung
5846006848301083051854302853545279775035881176911.

[0179] Beide Ordnungen sind Primzahlen.

[0180] Der Punkt
x = 4895927185759989738799640847700019479312599167010,
y = 4456609205960786928902978222286069176394833826606
liegt auf der elliptischen Kurve und erzeugt die volle Punktegruppe.

[0181] Zusammengefasst wurden oben Verfahren zur Transformation einer elliptischen Kurve beschrieben, welche es ermöglichen, ein skalares Vielfaches eines Punktes unter ausschließlicher Verwendung der x-Koordinate zu berechnen. Damit sind die erfindungsgemäßen Verfahren zur Skalar-Multiplikation gegen Timing-Angriffe geschützt.

[0182] Zusammengefasst ergeben sich für zwei Kurvenpunkte P1 und P2 einer elliptischen Kurve mit den jeweiligen x-Koordinaten x_1, x_2, die x-Koordinate x_3 des Punktes P3 = P1 + P2, die x-Koordinate x_4 des Punktes P4 = P1 – P2 und die x-Koordinate x_5 des Punktes P5 = 2·P1. In Abhängigkeit von der Charakteristik des Körpers ergeben sich zusammenfassend folgende Gleichungen:

1. Fall: Charakteristik des endlichen Körpers K = 2:

[0183] Die kurze Weierstraß-Gleichung einer nicht-supersingulären elliptischen Kurve hat die Form

$$y^2 + x\cdot y = x^3 + a_2\cdot x^2 + a_6. \tag{34}$$

Dann gilt:

$$(x_3 + x_4)(x_1 + x_2)^2 = x_1\cdot x_2, \tag{35}$$

und

$$x_5\cdot x_1^2 = x_1^4 + a_6. \tag{36}$$

2. Fall: Charakteristik des endlichen Körpers K ist 3:

[0184] Die kurze Weierstraß-Gleichung einer nicht-supersingulären elliptischen Kurve hat die Form

$$y^2 = x^3 + a_2\cdot x^2 + a_6. \tag{37}$$

[0185] Dann gilt:

$$(x_3 + x_4)(x_1 - x_2)^2 = a_6 + a_2\cdot x_1\cdot x_2 - x_1\cdot x_2\cdot(x_1 + x_2), \tag{38}$$

und

$$x_5(x_1^3 + a_2\cdot x_1^2 + a_6) = x_1^4 + a_6\cdot x_1 - a_2\cdot a_6. \tag{39}$$

3. Fall: Charakteristik des endlichen Körpers K größer als 3:

[0186] Die kurze Weierstraß-Gleichung einer nicht-supersingulären elliptischen Kurve hat die Form

$$y^2 = x^3 + a_4\cdot x + a_6. \tag{40}$$

[0187] Dann gilt:

$$(x_3 + x_4)(x_1 - x_2)^2 = 2\cdot(x_1 + x_2)\cdot(x_1\cdot x_2 + a_4) + 4\cdot a_6, \tag{41}$$

und

$$4 \cdot x_5 \cdot (x_1^3 + a_4 \cdot x_1 + a_6) = (a_1^2 - a_4)^2 - 8 \cdot a_6 \cdot x_1. \quad (42)$$

[0188] Die auf die oben beschriebene Weise ermittelte elliptische Kurve und die daraus resultierende und darauf durchgeführten, wie oben erläuterten, Gruppenoperationen wie Addition und skalare Multiplikation werden erfindungsgemäß in kryptographischen Verfahren eingesetzt, wie im Weiteren näher erläutert.

[0189] Gemäß den Ausführungsbeispielen werden im Rahmen der kryptographischen Verfahren lediglich die x-Koordinaten von Kurvenpunkten der ermittelten und verwendeten elliptischen Kurve übertragen, gespeichert und verarbeitet.

[0190] Jeder zu übertragende Kurvenpunkt einer elliptischen Kurve verringert zwar die Entropie des Schlüsselraumes von kryptographischen Schlüsseln um maximal ein Bit, bei allen derzeit praktisch relevanten kryptographischen Verfahren ist jedoch ein solcher Sicherheitsverlust vernachlässigbar und kann ohne nennenswerte Leistungsverluste durch eine Erhöhung der Schlüssellänge ausgeglichen werden.

[0191] Gemäß den dargelegten Ausführungsbeispielen werden Varianten von an sich bekannten, in der Literatur beschriebenen Kryptoverfahren erläutert, insbesondere des Diffie-Hellman-Verfahrens zur Schlüsseleinigung, des ElGamal-Public-Key-Verschlüsselungsverfahrens, des El-Gamal-Signaturverfahrens, des EC-DSA-Signaturverfahrens, des EC-GDSA-Signaturverfahrens, welche die gleiche Funktionalität wie die klassischen Verfahren bieten, aber ohne die Verwendung von y-Koordinaten der Kurvenpunkte der jeweiligen elliptischen Kurve auskommen.

[0192] Es ist darauf hinzuweisen, dass die Erfindung auch für alle anderen, vorzugsweise asymmetrischen Krypto-Verfahren ohne weiteres eingesetzt werden kann.

[0193] Ferner werden gemäß den folgenden Ausführungsbeispielen auf der Auswertung von Polynomen kleinen Grades beruhende Verifikationsverfahren für digitale Signaturen beschrieben, die ebenfalls vollständig auf die Verwendung der y-Koordinaten von Kurvenpunkten, welche auf der ausgewählten elliptischen Kurve liegen, verzichten. Diese werden für die verschiedenen Quellen der Charakteristiken der Grundkörper ausgeführt.

[0194] Anschaulich kann somit erfindungsgemäß bei Krypto-Verfahren auf Basis elliptischer Kurven über einem endlichen Körper vollständig auf die Speicherung, die Übertragung und Verarbeitung von y-Koordinaten von Kurvenpunkten, d.h. anders ausgedrückt, von Punkten auf der jeweils verwendeten ermittelten elliptischen Kurve, verzichtet werden.

[0195] Es werden für die Erzeugung und Verifizierung digitale Signaturen, für den Schlüsseltransport, Schlüsselaustausch, den Transport kryptographischer Schlüssel beim Austausch kryptographischer Schlüssel, asymmetrische Verschlüsslung und im Bereich der strengen Authentifikation Protokolle angegeben, welche ausschließlich die x-Koordinaten von Punkten elliptischer Kurven verwenden.

[0196] Es ist zu bemerken, dass aus protokolltechnischer Sicht keine Einschränkungen für die einzusetzenden Körper existieren.

[0197] Ferner werden Techniken zur Verifikation digitaler Signaturen auf Basis elliptischer Kurven beschrieben. Diese Techniken lassen sich auch in anderen kryptographischen Verfahren einsetzen.

[0198] Bei den im Weiteren beschriebenen kryptographischen Verfahren werden die oben beschriebenen Teil-Verfahren zum Berechnen der Multiplikation eines Punktes einer elliptischen Kurve mit einem Skalar als Unterroutine bzw. als Teil-Verfahren im Rahmen des jeweiligen Algorithmus verwendet.

[0199] Bei den unterschiedlichen im Weiteren beschriebenen Protokollen werden hauptsächlich 2 Aspekte berücksichtigt:

– die kryptographische Problemstellung wird so weit wie möglich durch ein Protokoll gelöst, das lediglich die Körperarithmetik zur Implementierung der Skalar-Multiplikation der elliptischen Kurven verwendet.

– es werden Varianten von Protokollen betrachtet, die nur die x-Koordinaten von Kurvenpunkten von elliptischen Kurven benötigen und berechnen müssen.

[0200] Die erfindungsgemäßen Verfahren gelten für beliebige, nicht-singuläre elliptische Kurven über endlichen Körpern beliebiger Charakteristik.

[0201] Zur einfacheren Darstellung der Ausführungsbeispiele werden die Verfahren am Beispiel von elliptischen Kurven über endlichen Körpern der Charakteristik 2 beschrieben. Selbstverständlich können alle im Weiteren beschriebenen Verfahren ohne weiteres auf ähnliche Körper anderer Charakteristiken übertragen werden, wie das weiter unten noch näher erläutert wird.

[0202] In einem kryptographischen Protokoll unter Verwendung elliptischer Kurven haben y-Koordinaten von Punkten einer elliptischen Kurve hauptsächlich folgende Funktionen:

– Testen, ob ein Punkt auf einer elliptischen Kurve liegt: Soll getestet werden ob ein Punkt (x, y) die Kurvengleichung einer elliptischen Kurve erfüllt oder ob ein gegebener Wert x die x-Koordinate eines gültigen Kurvenpunktes einer elliptischen Kurve sein kann, dann ist der Test leicht und effizient durchführbar, wenn die y-Koordinate gegeben ist. In diesem Fall müssen die beiden Koordinaten nur in die Kurvengleichung der elliptischen Kurve eingesetzt und die Gültigkeit der Gleichung überprüft werden. Falls die y-Koordinate nicht gegeben ist, müssen aufwendige Techniken zur Punktdekompression gemäß dem Stand der Technik verwendet werden.

Die erfindungsgemäß verwendeten elliptischen Kurven und Additionsformeln haben die Eigenschaft, dass ein beliebiges Körperelement stets x-Koordinate eines Punktes auf einer elliptischen Kurve ist, deren Gruppenstruktur entweder isomorph zur gewünschten elliptischen Kurve oder zu einer zugehörigen getwisteten elliptischen Kurve ist. Erfindungsgemäß werden daher nur elliptische Kurven verwendet, bei denen sowohl die Gruppenstruktur der gegebenen elliptischen Kurve als auch die der zugehörigen getwisteten elliptischen Kurve kryptographisch stark ist. Dadurch kann in den im Weiteren beschriebenen Protokollen auf Tests, ob ein Punkt tatsächlich auf einer elliptischen Kurve liegt, verzichtet werden, ohne die geheimen Schlüssel der Protokollteilnehmer zu kompromittieren. Aus diesem Grund werden erfindungsgemäß auch keine y-Koordinaten für solche Tests benötigt.

[0203] Kontrolle der Ordnung von Punkten:
Um mit den beispielsweise in [8] beschriebenen Additionsformeln sicherzustellen, dass ein von einem anderen Protokollteilnehmer übertragener Kurvenpunkt keine kleine Ordnung hat, sondern in der kryptographisch sicheren Untergruppe großer Ordnung liegt, ist die Kenntnis der y-Koordinate erforderlich.

– y-Koordinate als Teil einer übertragenen Nachricht: Die y-Koordinate ist erforderlich, um einen Punkt auf einer elliptischen Kurve eindeutig beschreiben zu können und kann daher relevante Information über eine Nachricht kodieren. Allerdings ist der Informationsgehalt der y-Koordinate auf ein Bit beschränkt und üblicherweise werden Verfahren zur Punktkompression eingesetzt, damit nicht eine lange Zahl als y-Koordinate übertragen werden muss.

Verfahren zur Dekompression von Punkten, die auf der x-Koordinate eines Kurvenpunktes einer elliptischen Kurve und dem Bit einer komprimierten y-Koordinate wieder den vollständigen Punkt berechnen, sind sehr rechenaufwendig und benötigen in etwa so viel Rechenzeit wie eine Skalar-Multiplikation eines Kurvenpunktes einer elliptischen Kurve. Die erfindungsgemäß dargelegten Protokolle verwenden jedoch keine y-Koordinaten. Dadurch ergeben sich Vereinfachungen der Protokolle und der zu übertragenen Nachrichten sowie ein Leistungsgewinn durch Vermeidung von Dekompressionsverfahren.

[0204] Alle im Weiteren beschriebenen kryptographischen Verfahren, insbesondere die im Weiteren beschriebenen asymmetrischen, das heißt Public-Key-Krypto-Verfahren, beruhen auf dem diskreten Logarithmus-Problem in der Punktgruppe einer elliptischen Kurve über einem endlichen Körper.

[0205] Ein Vorteil eines solchen Public-Key-Verfahrens liegt insbesondere darin, dass es einfach und effizient möglich ist, Schlüsselpaare mit einem geheimen Schlüssel und einem dem geheimen Schlüssel zugeordneten öffentlichen Schlüssel zu erzeugen.

[0206] Um ein Protokoll zur Schlüsselgenerierung zu implementieren, sind lediglich die Arithmetik des Grundkörpers und ein kryptographisch guter Zufallszahlen-Generator erforderlich.

[0207] Sei im Folgenden x_0 die x-Koordinate eines Kurvenpunktes P, der eine Untergruppe mit großer primer Ordnung d erzeugt.

[0208] Zur Schlüsselerzeugung wählt der Teilnehmer, das heißt der den Kurvenpunkt p erzeugende Prozessor eine Zufallszahl $0 < z < d$ und berechnet die x-Koordinate des Punktes $z \cdot P$. Die Zufallszahl z ist der geheime Schlüssel des Schlüsselpaares, und die berechnete x-Koordinate des Punktes $z \cdot P$ ist der öffentliche Schlüssel des Schlüsselpaares.

[0209] Diese Vorgehensweise ist im Folgenden in Form eines Pseudocodes anschaulich dargestellt:

1) $z \leftarrow$ random ();
2) $x \leftarrow z \cdot x_0$;

[0210] Für alle nachfolgend beschriebenen Protokolle ist es ausreichend, wenn nur die x-Koordinate des öffentlichen Schlüssels des jeweiligen verwendeten Schlüsselpaares bekannt ist.

[0211] Bei manchen im Weiteren beschriebenen Protokollen kann es erforderlich sein, dass der öffentliche Schlüssel nicht mittels $z \cdot x_0$, sondern mittels $z^{-1} \cdot x_0$ aus dem geheimen Schlüssel z berechnet wird. Der Wert z wird erfindungsgemäß dann modulo d invertiert, wozu eine weitere modulare Arithmetik erforderlich ist.

[0212] Fig. 2 zeigt ein Blockdiagramm, in dem eine erfindungsgemäße Variante des Diffie-Hellmann-Schlüsseleinigungsprotokolls 1 dargestellt ist. Um das an sich bekannte Diffie-Hellmann-Schlüsseleinigungsprotokoll erfindungsgemäß zu implementieren, sind lediglich die Arithmetik des Grundkörpers und ein kryptographisch guter Zufallszahlen-Generator erforderlich. Außerdem werden von den übertragenen und berechneten Punkten der elliptischen Kurven, die auf die oben beschriebene Weise ermittelt worden sind, nur die x-Koordinaten verwendet.

[0213] Fig. 2 zeigt in einem Blockdiagramm **200** einen ersten Rechner **201** sowie einen zweiten Rechner **202**, welche über ein Telekommunikationsnetz **203** miteinander gekoppelt sind.

[0214] Im Folgenden wird angenommen, dass E eine auf oben dargestellte Weise ermittelte elliptische Kurve ist derart, dass sowohl die Punktegruppe der elliptischen Kurve E als auch die Punktegruppe der zugehörigen getwisteten elliptischen Kurve kryptographisch geeignet sind.

[0215] Ferner sei x_0 die x-Koordinate eines Kurvenpunktes $P \in E$, der eine Untergruppe mit großer primer Ordnung d erzeugt und sei klein e das kleinste gemeinsame Vielfache der Kofaktoren von d, d.h. des großen Primteilers der Ordnung der elliptischen Kurve und d', das heißt des großen Primteilers der Ordnung der getwisteten elliptischen Kurve.

[0216] In einem ersten Schritt **204** wird von dem ersten Rechner **201** eine Zufallszahl z_A generiert.

[0217] Eine erste x-Koordinate x_A wird gebildet, indem die x-Koordinate x_0 des Kurvenpunkts P skalar multipliziert wird mit der gewählten Zufallszahl z_A (Schritt **205**).

[0218] Entsprechend wird von dem zweiten Rechner **202** ebenfalls eine zweite Zufallszahl z_B erzeugt (Schritt **206**).

[0219] Eine zweite x-Koordinate wird gebildet durch skalare Multiplikation der Zufallszahl mit der x-Koordinate x_0 eines Kurvenpunktes P (Schritt **207**).

[0220] Die erste x-Koordinate x_A wird in einer ersten Nachricht **208** von dem ersten Rechner **201** zu dem zweiten Rechner **202** übertragen.

[0221] Die zweite x-Koordinate x_B wird in einer zweiten Nachricht **209** von dem zweiten Rechner **202** zu dem ersten Rechner **201** übertragen.

[0222] Die von dem ersten Rechner **201** empfangene zweite x-Koordinate x_B – im Weiteren als empfangene zweite x-Koordinate y_A bezeichnet – wird in einem weiteren Schritt (Schritt **210**) skalar gemäß dem oben beschriebenen Verfahren mit dem kleinsten gemeinsamen Vielfachen der Kofaktoren von d und d', das heißt mit e, skalar multipliziert und für den Fall, dass y_A ungleich dem unendlich fernen Punkt O ist, wird die sich ergebende Koordinate anschließend mit der ersten Zufallszahl z_A multipliziert (Schritt **211**), womit von dem ersten Rechner **201** der Sitzungsschlüssel gebildet worden ist.

[0223] Die empfangene erste x-Koordinate x_A- im Weiteren als empfangene erste x-Koordinate y_B bezeichnet – wird von dem zweiten Rechner **202** in einem weiteren Schritt (Schritt **212**) mit dem kleinsten gemeinsamen Vielfachen e der Kofaktoren von d und d' multipliziert und für den Fall, dass das Ergebnis ungleich dem unendlich fernen Punkt O ist, wird das Ergebnis weiterhin mit der zweiten Zufallszahl z_B multipliziert (Schritt **213**), womit von dem zweiten Rechner **202** der Sitzungsschlüssel gebildet worden ist.

[0224] Durch die Schritte **210** und **212** wird verhindert, dass ein Teilnehmer bzw. ein Rechner **201**, **202** oder

ein externer Angreifer die ausgetauschten Kurvenpunkte durch Punkte mit kleiner Ordnung ersetzt. Auf diese Weise kann sichergestellt werden, dass der erzeugte Sitzungsschlüssel nicht kryptographisch schwach ist. Dieser Test verhindert aber nicht einen sogenannten „man-in-the-middle"-Angriff. Um einen solchen Angriff zu verhindern, sind erfindungsgemäß weitere Maßnahmen oder ein anderes Protokoll zur authentisierten Schlüsseleinigung vorgesehen, wie sie im Weiteren beschrieben sind.

[0225] Die sogenannten MTI-Protokolle (Matsumoto, Takashima, Imai-Protokolle) beschreiben eine Klasse von Protokollen zur authentisierten Schlüsseleinigung, die als Derivate des Diffie-Hellman-Protokolls ohne digitale Signaturen auskommen und damit unter Verwendung eines kryptographisch guten Zufallszahlen-Generators und des Charakteristik-2-Rechenwerks implementiert werden können.

[0226] Im Folgenden werden Varianten der ursprünglichen bekannten MTI-Protokolle auf Basis elliptischer Kurven beschrieben, wobei von übertragenen und berechneten Punkten wiederum nur die x-Koordinaten verwendet werden.

MTI/A0-Protokoll

[0227] Im Folgenden wird das erfindungsgemäß angepasste MTI/A0-Protokoll beschrieben, welches eine auf die oben beschriebene Weise ermittelte elliptische Kurve verwendet.

[0228] Anders ausgedrückt wird gemäß diesem Ausführungsbeispiel vorausgesetzt, dass die elliptische Kurve E eine elliptische Kurve ist, so dass sowohl die Punktegruppe der elliptischen Kurve als auch die Punktegruppe der zugehörigen getwisteten elliptischen Kurve kryptographisch geeignet sind, das heißt die oben beschriebenen kryptographischen Gütekriterien erfüllen.

[0229] Wiederum sei x_0 die x-Koordinate eines Kurvenpunktes $P \in E$. der einer Untergruppe mit großer primer Ordnung d erzeugt, sei e das kleinste gemeinsame Vielfache der Kofaktoren von d, das heißt des großen Primteilers der Ordnung der elliptischen Kurve, und d', das heißt des großen Primteilers der Ordnung der zugehörigen getwisteten elliptischen Kurve, und sei p_A der geheime Schlüssel und $Q_A = x(p_A*P)$ der öffentliche Schlüssel des Benutzers des ersten Rechners **301** bzw. p_B der geheime Schlüssel und $Q_B = x(p_B*B)$ der öffentliche Schlüssel des Benutzers des zweiten Rechners **302** (vergleiche Blockdiagramm **300** in Fig. 3).

[0230] Der erste Rechner **301** und ein zweiter Rechner **302** sind wiederum über ein Kommunikationsnetz **303** vorzugsweise das Internet, miteinander gekoppelt.

[0231] In einem ersten Schritt (Schritt **304**) wird von dem ersten Rechner **301** das Zertifikat des öffentlichen Schlüssels Q_B des Benutzers des zweiten Rechners **302** überprüft und für den Fall, das dass Zertifikat ungültig ist, wird eine Fehlermeldung ausgegeben. Ist das Zertifikat des öffentlichen Schlüssels Q_B des Benutzers des zweiten Rechners **302** gültig, so wird in einem weiteren Schritt (Schritt **305**) eine erste Zufallszahl $z_A > 0$ erzeugt. Eine erste x-Koordinate x_A wird in einem weiteren Schritt (Schritt **306**) ermittelt, indem die x-Koordinate x_0 des Kurvenpunktes P skalar multipliziert wird mit der ersten Zufallszahl z_A.

[0232] Von dem zweiten Rechner **302** wird seinerseits das Zertifikat des öffentlichen Schlüssels Q_A des Benutzers des ersten Rechners **301** überprüft und bei dessen Ungültigkeit wird eine Fehlermeldung ausgegeben (Schritt **307**).

[0233] Ist das Zertifikat des öffentlichen Schlüssels Q_A des Benutzers des ersten Rechners **301** gültig, so wird von dem zweiten Rechner **302** unter Verwendung des Zufallszahlen-Generators eine zweite Zufallszahl z_B erzeugt (Schritt **308**) und es wird eine zweite x-Koordinate x_B ermittelt (Schritt **309**), indem die x-Koordinate x_0 des Kurvenpunktes P skalar multipliziert wird mit der zweiten Zufallszahl z_B.

[0234] Die erste x-Koordinate x_A wird in einer ersten digitalen Nachricht **310** von dem ersten Rechner **301** zu dem zweiten Rechner **302** übertragen.

[0235] Die zweite x-Koordinate x_B wird in einer zweiten digitalen Nachricht **311** von dem zweiten Rechner **302** zu dem ersten Rechner **301** übertragen.

[0236] Die in der zweiten Nachricht **311** von dem ersten Rechner **301** empfangene erste x-Koordinate x_B, im Weiteren als y_A bezeichnet, wird von dem ersten Rechner **301** mit dem kleinsten gemeinsamen Vielfachen e der Kofaktoren von d und d' multipliziert und für den Fall, dass der sich ergebende Wert gleich dem unendlich

fernen Punkt O ist, wird eine Fehlermeldung ausgegeben (Schritt **312**).

[0237] Ist der sich ergebende Wert ungleich dem unendlich fernen Punkt O, so wird der empfangene Wert y_A multipliziert mit dem geheimen Schlüssel p_A des Benutzers des ersten Rechners **301**, womit sich ein erster Zwischenwert t_A ergibt (Schritt **313**).

[0238] Ferner wird von dem ersten Rechner **301** der öffentliche Schlüssel Q_B des Benutzers des zweiten Rechners **302** mit der ersten Zufallszahl z_A multipliziert, womit ein zweiter Zwischenwert s_A gebildet wird (Schritt **314**).

[0239] Ferner verknüpft der erste Rechner **301** den ersten Zwischenwert t_A mit dem zweiten Zwischenwert s_A kommutativ, womit der Sitzungsschlüssel k_A von dem ersten Rechner **301** gebildet wird (Schritt **315**).

[0240] Der zweite Rechner **302** multipliziert die in der ersten Nachricht **309** enthaltene und empfangene erste x-Koordinate x_A, im Weiteren bezeichnet als y_B, mit dem kleinsten gemeinsamen Vielfachen e der Kofaktoren von d und d' und erzeugt für den Fall, dass das Ergebnis der Multiplikation gleich dem unendlich entfernten Punkt O ist, ein Fehlersignal (Schritt **316**).

[0241] Ist das Ergebnis ungleich dem unendlich fernen Punkt O, so wird von dem zweiten Rechner **302** die erste x-Koordinate y_B multipliziert mit dem geheimen Schlüssel p_B des Benutzers des zweiten Rechner **302**, womit ein dritter Zwischenwert t_B gebildet wird (Schritt **317**).

[0242] Ferner wird der öffentliche Schlüssel Q_A des Benutzers des ersten Rechners **301** mit der zweiten Zufallszahl z_B, welche lediglich dem zweiten Rechner **302** bekannt ist, skalar multipliziert, womit ein vierter Zwischenwert s_B berechnet wird (Schritt **318**).

[0243] In einem weiteren Schritt (Schritt **319**) wird unter Verwendung der gleichen kommutativen Verknüpfung, welche in dem ersten Rechner **301** durchgeführt wird, eine kommutative Verknüpfung gebildet zwischen dem dritten Zwischenwert t_B und dem vierten Zwischenwert s_B, womit von dem zweiten Rechner **302** der Sitzungsschlüssel kg gebildet wird, welcher gleich ist dem von dem ersten Rechner **301** gebildeten Sitzungsschlüssel k_A.

[0244] Nach Ausführung des in Fig. 3 dargestellten Protokolls besitzen sowohl der erste Rechner **301** als auch der zweite Rechner **302** den Sitzungsschlüssel $z_A \cdot p_B \circ z_B \cdot p_A$. Als kommutative Verknüpfung können beispielsweise die normale Addition oder Multiplikation im Grundkörper als Operation verwendet werden.

[0245] Durch die Schritte **312** und **316** wird verhindert, dass ein Teilnehmer, das heißt der erste Rechner **301**, der zweite Rechner **302** oder auch ein externer Angreifer die ausgetauschten Kurvenpunkte der elliptischen Kurven durch Punkte mit kleiner Ordnung ersetzt. Auf diese Weise kann sichergestellt werden, dass der generierte Sitzungsschlüssel nicht kryptographisch schwach ist.

[0246] Zur weiteren Erhöhung der kryptographischen Sicherheit ist es vorgesehen, nicht den auf oben beschriebene Weise unmittelbar berechneten Schlüssel $k_A = k_B$ als Sitzungsschlüssel zu verwenden, sondern ein mittels einer Schlüssel-Ableitungsfunktion f(Key-Derivation Function) berechneten Wert $f(k_A) = f(k_B)$.

[0247] Die Schlüssel-Ableitungsfunktion f sollte eine Funktion sein, welche die algebraische Funktion des endlichen Körpers zerstört.

[0248] Fig. 4 zeigt in einem Blockdiagramm **400** die einzelnen Schritte gemäß dem MTI/B0-Protokoll unter Verwendung einer erfindungsgemäß ermittelten elliptischen Kurve.

[0249] Die Verfahrensschritte bis zum Austausch der x-Koordinaten zwischen dem ersten Rechner **301** und dem zweiten Rechner **302** sind gleich der Vorgehensweise wie gemäß dem MTI/A0-Protikoll und werden aus diesem Grund nicht erneut erläutert.

[0250] Gemäß diesem Ausführungsbeispiel werden jedoch die jeweils empfangenen x-Koordinaten y_A, y_B in anderer Weise weiterverarbeitet.

[0251] Gemäß diesem Ausführungsbeispiel wird von dem ersten Rechner **302** die zweite x-Koordinate y_A mit dem multiplikativen Inversen p_A^{-1} des geheimen Schlüssels p_A des Benutzers des ersten Rechners **301** multip-

liziert (Schritt **401**), womit der erste Zwischenwert t_A gebildet wird. (Das Inverse p_A^{-1} wird in Z/dZ berechnet).

[0252] Ferner wird von dem ersten Rechner **301** ein zweiter Zwischenwert S_A berechnet mittels Multiplikation der x-Koordinate x_0 des Kurvenpunktes P mit der ersten Zufallszahl z_A (Schritt **402**).

[0253] Der Sitzungsschlüssel k_A wird in einem weiteren Schritt (Schritt **403**) von dem ersten Rechner **301** durch kommutative Verknüpfung des ersten Zwischenwertes t_A mit dem zweiten Zwischenwert s_A gebildet.

[0254] Von dem zweiten Rechner **302** wird ein dritter Zwischenwert t_B gebildet mittels Multiplikation der empfangenen ersten x-Koordinate y_B mit dem multiplikativen Inversen p_B^{-1} des geheimen Schlüssels p_B des Benutzers des zweiten Rechners **302** (Schritt **404**).

[0255] Ein vierter Zwischenwert s_B wird gebildet mittels Multiplikation der x-Koordinate x_0 des Kurvenpunktes P mit der zweiten Zufallszahl z_B (Schritt **405**).

[0256] Der Sitzungsschlüssel k_B wird von dem zweiten Rechner **302** mittels kommutativer Verknüpfung des dritten Zwischenwertes t_B mit dem vierten Zwischenwert s_B gebildet (Schritt **406**).

[0257] Nach Durchführung des MTI/B0-Protokolls besitzen beide Rechner **301**, **302** den Sitzungsschlüssel $z_A \cdot x_0 \circ z_B \cdot x_0$.

[0258] Auch in diesem Fall kann wiederum unter Verwendung einer Schlüssel-Ableitungsfunktion f der eigentlich zu verwendende Sitzungsschlüssel gebildet werden.

[0259] Die multiplikativen Inversen p_A^{-1}, p_B^{-1} der geheimen Schlüssel der Benutzer des ersten Rechners **301** und des zweiten Rechners **302** können bereits bei der Generierung der Schlüssel beziehungsweise der Zertifikate vorberechnet und in dem jeweiligen Rechner **301**, **302** gespeichert sein.

[0260] Fig. 5 zeigt in einem Blockdiagramm **500** eine erfindungsgemäße Realisierung einer Variante des MTI/C0-Protokolls.

[0261] Wiederum sind die Verfahrensschritte bis zum Austausch der x-Koordinaten x_A, x_B mit dem Vorgehen im Rahmen des MTI/A0-Protokolls identisch und werden aus diesem Grunde nicht erneut erläutert.

[0262] Der erste Rechner **301** multipliziert die empfangene zweite x-Koordinate y_A mit dem multiplikativen Inversen p_A^{-1} des geheimen Schlüssels p_A des Benutzers des ersten Rechners **301** (Schritt **501**) und multipliziert das Ergebnis mit der ersten Zufallszahl z_A (Schritt **502**), womit der Sitzungsschlüssel k_A gebildet wird.

[0263] Der zweite Rechner **302** multipliziert die empfangene erste x-Koordinate y_B mit dem multiplikativen Inversen p_B^{-1} des geheimen Schlüssels p_B des Benutzers des zweiten Rechners **302** (Schritt **503**) und multipliziert anschließend das Ergebnis mit der zweiten Zufallszahl z_B (Schritt **504**), womit von dem zweiten Rechner **302** der Sitzungsschlüssel kg gebildet wird.

[0264] Nach Ausführung des MTI/C0-Protokolls besitzen beide Rechner **301**, **302** den Sitzungsschlüssel $z_A \cdot z_B \cdot x_0$. Wiederum gelten sinngemäß die gleichen Anmerkungen wie bezüglich dem Ausführungsbeispiel betreffend das MTI/A0-Protokoll. Wiederum können die multiplikativen Inversen der jeweiligen geheimen Schlüssel bereits bei der Generierung der geheimen Schlüssel beziehungsweise der Zertifikate vorberechnet und gespeichert werden.

[0265] Fig. 6 zeigt in einem Blockdiagramm **600** die erfindungsgemäße Implementierung einer Variante des sogenannten MTI/C1-Protokolls, bei dem lediglich x-Koordinaten verwendet werden.

[0266] Die Schritte **304**, **305**, **306**, welche von dem ersten Rechner **301** ausgeführt werden und sowie die Schritte **307**, **308**, **309**, welche von dem zweiten Rechner **302** ausgeführt werden, sind mit dem Ausführungsbeispiel gemäß dem MTI/A0-Protokoll identisch und werden aus diesem Grund nicht erneut erläutert.

[0267] Im Unterschied zu den oben beschriebenen Protokollen wird gemäß diesem Ausführungsbeispiel die erste x-Koordinate x_A noch mit dem geheimen Schlüssel p_A des Benutzers des ersten Rechners **301** multipliziert (Schritt **601**) und erst anschließend wird das Produkt in einer ersten digitalen Nachricht **602** zu dem zweiten Rechner **302** übertragen.

[0268] In entsprechender Weise wird von dem zweiten Rechner **302** die zweite x-Koordinate x_B mit dem geheimen Schlüssel p_B des Benutzers des zweiten Rechners **302** multipliziert (Schritt **603**) und nur das Ergebnis wird in einer zweiten digitalen Nachricht **604** von dem zweiten Rechner **302** zu dem ersten Rechner **301** übertragen.

[0269] Ist das Produkt des von dem ersten Rechner **301** empfangenen Wertes – im Weiteren als y_A bezeichnet – in der zweiten Nachricht **604** mit dem kleinsten gemeinsamen Vielfachen e gleich dem unendlich fernen Punkt O, dann erzeugt der erste Rechner **301** eine Fehlermeldung (Schritt **605**), sonst wird der empfangene Wert mit der ersten Zufallszahl z_A multipliziert (Schritt **606**), womit der Sitzungsschlüssel k_A von dem ersten Rechner **301** erzeugt wird.

[0270] In entsprechender Weise wird von dem zweiten Rechner **302** dessen in der ersten Nachricht **602** enthaltener und von dem zweiten Rechner **302** empfangener Wert – im Weiteren bezeichnet mit y_B – mit dem kleinsten gemeinsamen Vielfachen e multipliziert und es wird geprüft, ob das Produkt gleich ist dem fernen Punkt O.

[0271] Ist dies der Fall, so wird eine Fehlernachricht erzeugt (Schritt **607**), sonst wird der empfangene Wert mit der zweiten Zufallszahl z_B multipliziert, womit der Sitzungsschlüssel k_B von dem zweiten Rechner **302** ermittelt wird (Schritt **608**).

[0272] Nach Ausführung dieses Protokolls besitzen der erste Rechner **301** und der zweite Rechner **302** beide den Sitzungsschlüssel $z_A \cdot z_B \cdot p_A \cdot p_B \cdot x_0$. Es gelten wiederum die gleichen obigen Anmerkungen zu Ausgestaltungen dieses Protokolls.

[0273] Fig. 7 zeigt an einem Blockdiagramm **700** die erfindungsgemäße Implementierung einer Verschlüsselung beziehungsweise Entschlüsselung von digitalen Daten gemäß einer erfindungsgemäßen Ausgestaltung des Verfahrens gemäß ElGamal. Hierzu sind erfindungsgemäß zum Verschlüsseln die Arithmetik des Grundkörpers und ein kryptographisch guter Zufallszahlen-Generator in einem ersten Rechner **701** und zum Entschlüsseln lediglich die Arithmetik des Grundkörpers in einem zweiten Rechner **702** erforderlich. Der erste Rechner **701** und der zweite Rechner **702** sind über ein Telekommunikationsnetz **703** miteinander gekoppelt.

[0274] Außerdem werden von den übertragenen und berechneten Punkten der elliptischen Kurve wiederum nur die x-Koordinaten verwendet.

[0275] Sei im Folgendem x_0 die x-Koordinate eines Kurvenpunkts P, der eine Untergruppe mit großer primer Ordnung d erzeugt, seien p_B der geheime Schlüssel und $Q_B = x(p_B \cdot P)$ der öffentliche Schlüssel des Benutzers des zweiten Rechners **702**, seien p_A der geheime Schlüssel und $Q_A = x(p_A \cdot P)$ der öffentliche Schlüssel des Benutzers des ersten Rechners **701**.

[0276] Damit der erste Rechner **701** eine Nachricht $0 \leq m \leq d$ für den zweiten Rechner **702** verschlüsseln kann, werden folgende Verfahrensschritte durchgeführt.

[0277] In einem ersten Schritt wird eine erste Zufallszahl z_A erzeugt (Schritt **704**). Anschließend wird die x-Koordinate x_0 des Kurvenpunktes P mit der ersten Zufallszahl z_A multipliziert (Schritt **705**) und es wird der öffentliche Schlüssel Q_B des Benutzers des zweiten Rechners **702** ebenfalls mit der ersten Zufallszahl z_A multipliziert, womit ein erster Zwischenwert y_A gebildet wird (Schritt **706**).

[0278] Eine verschlüsselte Nachricht **708** wird gebildet (Schritt **707**) unter Verwendung des in Schritt **705** gebildeten ersten Zwischenwerts y_A, der beispielsweise der jeweiligen Nachricht hinzuaddiert wird und unter Verwendung der ersten x-Koordinate x_A, welche in Schritt **705** erzeugt wurde als für die Verschlüsselung verwendeter Schlüssel.

[0279] Die verschlüsselte Nachricht **708** wird von dem zweiten Rechner **702** nach dessen Empfang entschlüsselt, indem der empfangene Wert x_B multipliziert wird mit dem geheimen Schlüssel p_B des Benutzers des zweiten Rechners **702** (Schritt **709**), und es erfolgt beispielsweise das Bilden der Differenz aus dem in Schritt **709** ermittelten Wert und dem mitübertragenen Wert y_B (Schritt **710**).

[0280] Sämtliche arithmetischen Schritte werden im Grundkörper ausgeführt.

[0281] Üblicherweise sollte der Benutzer des zweiten Rechners **702** noch ein Zertifikat seines öffentlichen

Schlüssels Q_B besitzen, welches der erste Rechner **701** vor der Verwendung des öffentlichen Schlüssels Q_B des Benutzers des zweiten Rechners **702** prüfen sollte.

[0282] Eine solche Prüfung eines Zertifikates kann beispielsweise mittels der im Weiteren beschriebenen Verfahren zur Überprüfung digitaler Signaturen erfolgen.

[0283] Falls in einer Anwendung die zu verschlüsselnde Nachricht m ein Element eines endlichen Körpers ist, kann es vorteilhaft sein, wenn nicht die x-Koordinate des Punktes y_A, sondern ein von y_A abgeleiteter Wert $f(y_A)$ zum Verschlüsseln der zu verschlüsselnden Nachricht m verwendet wird.

[0284] Es sollte f eine Funktion sein, welche die algebraische Struktur des endlichen Körpers zerstört, vorzugsweise eine Schlüssel-Ableitungsfunktion oder eine Hash-Funktion.

[0285] Auf diese Weise kann einerseits Angriffen auf die zu verschlüsselnde Nachricht m selbst, welche die algebraische Struktur ausnutzen vorgebeugt werden, andererseits können statistische Schwächen der einzelnen Bits von y_A ausgeglichen werden.

[0286] Eine Vorgehensweise zur Konstruktion von Authentisierungsprotokollen mittels Public-Key-Verfahren besteht darin, die Verschlüsselungsverfahren wie oben beschrieben mit symmetrischen Verschlüsselungsverfahren zu kombinieren.

[0287] Auf diese Weise erhält man ein 2-Runden-Challenge-Response Protokoll zur Nachrichtenauthentisierung, das auf asymmetrischen Krypto-Verfahren beruht.

[0288] Um ein solches Authentisierungsverfahren zu implementieren, sind für den zweiten, die Authentisierung durchführenden, Rechner **802** die Arithmetik des Grundkörpers, ein kryptographisch guter Zufallszahlen-Generator und das verwendete symmetrische Verschlüsselungsverfahren notwendig.

[0289] Ein erster Rechner **801**, welcher zu authentisieren ist, benötigt hingegen die Arithmetik des Grundkörpers und das verwendete symmetrische Verschlüsselungsverfahren.

[0290] Der erste Rechner **801** und der zweite Rechner **802** sind über ein Telekommunikationsnetzwerk **803** miteinander gekoppelt.

[0291] Außerdem ist anzumerken, dass von den übertragenen berechneten Punkten der elliptischen Kurven wiederum nur die x-Koordinaten erfindungsgemäß verwendet werden.

[0292] Sei im Folgendem x_0 die x-Koordinate eines Kurvenpunktes P, der eine Untergruppe mit großer primer Ordnung d erzeugt, seien p_P der geheime Schlüssel des Benutzers des ersten Rechners **801** und $Q_P = x(p_P*P)$ der öffentliche Schlüssel des Benutzers des ersten Rechners **801**. Sei MAC(k, m) ein (symmetrisches) Verfahren zur Erzeugung eines Message Authentication Codes (MACs) der Nachricht m unter Verwendung des Schlüssels k.

[0293] Der Protokollablauf des Authentisierungsverfahrens gemäß diesem Ausgangsbeispiel der Erfindung ist in dem Blockdiagramm **800** in Fig. 8 dargestellt.

[0294] Der zweite Rechner **802** prüft das Zertifikat des öffentlichen Schlüssel Q_P des ersten Rechners **801** und gibt eine Fehlermeldung aus, wenn das Zertifikat ungültig ist (Schritt **804**).

[0295] In einem weiteren Schritt (Schritt **805**) erzeugt der zweite Rechner **802** eine erste Zufallszahl z_1 und anschließend erzeugt der zweite Rechner **802** in einem weiteren Schritt (Schritt **806**) eine zweite Zufallszahl z_2.

[0296] Es wird ein erster Zwischenwert x_V mittels Multiplikation der x-Koordinate x_0 des Kurvenpunktes P mit der ersten Zufallszahl z_1 ermittelt (Schritt **807**) sowie eine zweiter Zwischenwert y_V mittels Multiplikation des öffentlichen Schlüssels Q_P des ersten Rechners **801** mit der ersten Zufallszahl z_1 (Schritt **808**).

[0297] In einer ersten digitalen Nachricht **809** werden der ersten Zwischenwert x_V sowie die Summe des zweiten Zwischenwertes y_V und der zweiten Zufallszahl z_2 zu dem ersten Rechner **801** übertragen.

[0298] Die von dem ersten Rechner **801** empfangenen zwei Werte (x_V, $y_v + z_2$) – im Weiteren bezeichnet als

Wertetupel (x_P, y_P) werden von dem ersten Rechner **801** derart bearbeitet, dass der erste empfangene Wert x_P, welcher gleich ist dem ersten Zwischenwert x_V, multipliziert wird mit dem geheimen Schlüssel p_P des Benutzers des ersten Rechners **801**, womit ein dritter Zwischenwert gebildet wird (Schritt **810**).

[0299] Ferner wird ein MAC über die Nachricht m unter Verwendung der Differenz des ersten empfangenen Werts x_P und des zweiten empfangenen Werts y_P als Schlüssel ermittelt, womit ein Authentisierungswert r bestimmt wird (Schritt **811**), der in einer zweiten digitalen Nachricht **812** zu dem zweiten Rechner **802** übermittelt wird.

[0300] Der zweite Rechner **802** bildet einen MAC über die zu authentisierende Nachricht m unter Verwendung der zweiten Zufallszahl z_2 als kryptographischen Schlüssel (Schritt **813**), womit ein vierter Zwischenwert t gebildet wird.

[0301] Das Ergebnis wird mit dem empfangenen Authentisierungswert r verglichen und bei Übereinstimmung der beiden Werte wird die Authentisierung als erfolgreich angesehen (Schritt **814**).

[0302] Bei der Implementierung von Verfahren zur Erzeugung digitaler Signaturen ist zu berücksichtigen, dass alle gebräuchlichen Verfahren zur digitalen Signatur zusätzlich zur Arithmetik des Grundkörpers und den darauf basierenden elliptischen Kurven auch modulare Berechnungen, zumindest Berechnungen über Z, benötigen.

[0303] Im Gegensatz dazu ist zur Implementierung der gebräuchlichen Verfahren zur Überprüfung digitaler Signaturen auf Basis elliptischer Kurven (ElGamal, EC-DAS, EC-GDSA) bereits die Arithmetik des Grundkörpers ausreichend.

[0304] In den nachfolgenden Beschreibungen der Ausführungsbeispiele zum Erstellen und Überprüfen digitaler Signaturen wird ein Signaturwert r einer Signatur (r, s) in verschiedenen mathematischen Zusammenhängen verwendet, ohne dass dieses durch die Notation explizit angegeben wird.

[0305] Der Wert n wird entweder als Körperelement des endlichen Körpers $GF(2^m)$ bzw. in diesem Kontext auch als die x-Koordinate eines Punktes auf einer elliptischen Kurve, oder als natürliche Zahl (bei Berechnungen modulo der Ordnung des Basispunktes oder als Skalar bei Skalar-Multiplikationen) interpretiert.

[0306] Es wird jeweils eindeutig aus dem Kontext der arithmetischen Operationen deutlich, wie der Wert r jeweils zu interpretieren ist.

[0307] Um EC-ElGamal-Signaturen zu erzeugen, sind die Arithmetik des Grundkörpers, ein kryptographisch guter Zufallszahlen-Generator und eine Arithmetik zum Rechnen in Z/dZ, wobei d die Ordnung der Punktegruppe der elliptischen Kurve ist, erforderlich.

[0308] Bei der im Folgenden beschriebenen Protokoll-Variante werden von den berechneten Punkten der elliptischen Kurve wiederum nur die x-Koordinaten verwendet.

[0309] Sei im Folgenden E eine elliptische Kurve, so dass die Punktegruppe der Kurve kryptographisch geeignet ist, sei x_0 die x-Koordinate eines Kurvenpunkts P, der eine Untergruppe mit großer primer Ordnung d erzeugt und seien p_A der geheime Schlüssel und $Q_A = x(p_A*P)$ der öffentliche Schlüssel des Benutzers des ersten Rechners.

[0310] Damit der erste Rechner den Hash-Wert $0 \leq h(m) < d$ einer Nachricht m signieren kann, sind folgende, in dem Ablauf-Diagramm **900** in Fig. 9 dargestellten Verfahrensschritte notwendig.

[0311] In einem ersten Schritt wird eine Zufallszahl z erzeugt, wobei gilt: $0 < z < d$ (Schritt **901**). Die x-Koordinate x_0 des Kurvenpunktes P einer elliptischen Kurve E wird mit der Zufallszahl z multipliziert, womit ein erster Zwischenwert r gebildet wird (Schritt **902**).

[0312] Ferner wird die multiplikative Inverse der Zufallszahl z^{-1} modulo d gebildet und als zweiter Zwischenwert z ermittelt (Schritt **903**). Anschließend wird das Ergebnis multipliziert mit der Differenz des Hash-Werts h(m) und dem Produkt des ersten Zwischenwerts r mit dem geheimen Schlüssel p_A des Benutzers des ersten Rechners, wobei das Produkt modulo d berechnet wird (Schritt **904**), womit ein dritter Zwischenwert s gebildet wird.

[0313] Es ist anzumerken, dass die Schritte **903** und **904** in dem Körper Z/dZ berechnet werden.

[0314] Ist der dritte Zwischenwert s = 0, dann wird das Verfahren erneut durchgeführt für eine neue Zufallszahl beginnend in Schritt **901**.

[0315] Sonst werden der erste Zwischenwert r und der dritte Zwischenwert s als Ergebnis ausgegeben. Das Ergebnis, d.h. das Zahlenpaar (r, s) stellt eine gültige Signatur des Hash-Werts h(m) der Nachricht m dar.

[0316] Fig. 10 zeigt ein Verfahren zur Verifikation von ElGamal-Signaturen, wobei wiederum nur die x-Koordinaten verwendet werden.

[0317] Um das Protokoll zur Verifikation von EC-ElGamal-Signaturen zu implementieren, ist lediglich die Arithmetik des Grundkörpers notwendig.

[0318] Sei im Folgenden E eine elliptische Kurve über $GF(2^m)$mit der Kurvengleichung

$$y^2 + x \cdot y = x^3 + a_2 \cdot x^2 + a_6, \tag{43}$$

so dass die Punktegruppe der elliptischen Kurve kryptographisch geeignet ist, sei x_0 die x-Koordinate eines Kurvenpunktes P, der eine Untergruppe mit großer primer Ordnung d erzeugt und sei $Q_A = x(p_A * P)$ der öffentliche Schlüssel des Benutzers des ersten Rechners.

[0319] Damit der zweite Rechner eine Signatur (r, s) einer Nachricht m mit Hash-Wert $0 \leq h(m) < d$ verifizieren kann, werden folgende, in dem Ablauf-Diagramm **1000** in Fig. 10 dargestellten Verfahrensschritte von dem zweiten Rechner durchgeführt.

[0320] Zunächst wird das Zertifikat des öffentlichen Schlüssels Q_A des Benutzers des ersten Rechners überprüft und bei Ungültigkeit des Zertifikats wird eine Fehlermeldung ausgegeben (Schritt **1001**).

[0321] Anschließend wird geprüft, ob der Wert r der Signatur $(r, s) \in GF(2^m)$ ist, ob der Wert $r \neq 0$ ist, sowie ob der Wert s größer als 0 und kleiner als d ist.

[0322] Ist eine der Bedingungen nicht erfüllt, so wird wiederum eine Fehlermeldung ausgegeben (Schritt **1002**).

[0323] Sonst wird der öffentliche Schlüssel Q_A des Benutzers des ersten Rechners mit dem ersten Wert r der Signatur (r, s) multipliziert, womit ein vierter Zwischenwert x gebildet wird (Schritt **1003**).

[0324] In einem weiteren Schritt (Schritt **1004**) wird ein fünfter Zwischenwert y mittels Multiplikation des ersten Werts r mit dem zweiten Wert s der Signatur (r, s) gebildet.

[0325] Ein sechster Zwischenwert z wird gebildet mittels Multiplikation des Hash-Werts h(m) mit der x-Koordinate x_0 des Punktes P (Schritt **1005**).

[0326] Wenn ein quadratisches Polynom über dem vierten Zwischenwert x, dem fünften Zwischenwert y, sowie dem sechsten Zwischenwert z gleich 0 ist, dann gilt die Signatur als verifiziert (Schritt **1006**).

[0327] Die Schritte **1003**, **1004**, **1005** sind Skalar-Multiplikationen mit Punkten der elliptischen Kurve E. In dem Schritt **1006** wird anschaulich ein quadratisches Polynom über dem endlichen Körper $GF(2^m)$ ausgewertet. Das heißt, dass zur Berechnung des Werts q(x, y, z) lediglich die Arithmetik des Körpers GF(2m) notwendig ist.

[0328] Dabei hat das Polynom $q(x_1, x_2, x_3)$ die Form:

$$q(x_1, x_2, x_3) = x_3^2(x_1 + x_2)^2 + x_3 x_1 x_2 + x_1^2 x_2^2 + a_6, \tag{44}$$

wobei a_6 der entsprechende Parameter der elliptischen Kurve E ist.

[0329] Um EC-DSA-Signaturen erzeugen zu können, sind die Arithmetik des Grundkörpers, ein kryptographisch guter Zufallszahlen-Generator und eine Arithmetik zum Rechnen in Z/dZ, wobei d die Ordnung der

Punktegruppe der elliptischen Kurve E ist, erforderlich.

[0330] Bei der im Folgenden beschriebenen Protokollvariante werden wiederum von den berechneten Punkten der elliptischen Kurve E nur die x-Koordinaten verwendet.

[0331] Sei im folgenden E eine elliptische Kurve, so dass die Punktegruppe der elliptischen Kurve kryptographisch geeignet ist, sei x_0 die x-Koordinate eines Kurvenpunktes P, der eine Untergruppe mit großer primer Ordnung d erzeugt und sei $p_A = x(p_A*P)$ der öffentliche Schlüssel des Benutzers des ersten Rechners.

[0332] Damit der erste Rechner den Hash-Wert $0 < h(m) < d$ einer Nachricht m signieren kann, sind folgende Schritte vorgesehen, welche in dem Ablaufdiagramm **1100** in Fig. 11 dargestellt sind und von dem ersten Rechner durchgeführt werden.

[0333] Zunächst wird eine Zufallszahl z gebildet, für die gilt $0 < z < d$ (Schritt **1101**).

[0334] Anschließend wird die x-Koordinate x_0 des Kurvenpunktes P mit der Zufallszahl z multipliziert, womit ein erster Zwischenwert r gebildet wird (Schritt **1102**).

[0335] Anschließend wird ein zweiter Zwischenwert gebildet, indem die multiplikative Inverse der Zufallszahl z modulo d gebildet wird (Schritt **1103**).

[0336] Ein dritter Zwischenwert s wird gebildet durch Multiplikation modulo d des zweiten Zwischenwerts mit der Summe des Hash-Werts h(m) und dem Produkt des ersten Zwischenwerts r mit dem geheimen Schlüssel p_A des Benutzers des ersten Rechners (Schritt **1104**).

[0337] Ist der dritte Zwischenwert gleich dem Wert 0, dann wird das Verfahren mit Schritt **1101** von vorne begonnen.

[0338] Sonst wird das Zahlenpaar mit dem ersten Zwischenwert und dem dritten Zwischenwert ausgegeben, wobei das Zahlenpaar (r, s) eine gültige Signatur des Hash-Wertes h(m) der Nachricht m darstellt.

[0339] Es ist anzumerken, dass die Verfahrensschritte **1103** und **1104** jeweils in dem Körper Z/dZ berechnet werden.

[0340] Der Algorithmus zur Verifikation von EC-DSA-Signaturen unterscheidet sich wiederum von den bekannten Verfahren, da erfindungsgemäß nur die x-Koordinaten verwendet werden.

[0341] Um das Protokoll zur Verifikation von EC-DSA-Signaturen zu implementieren, ist erfindungsgemäß lediglich die Arithmetik des Grundkörpers erforderlich.

[0342] Sei im Folgendem E eine elliptische Kurve über $GF(2^m)$ mit der Kurvengleichung

$$y^2 + x \cdot y = x^3 + a_2 \cdot x^2 + a_6, \tag{45}$$

so dass die Punktegruppe der elliptischen Kurve kryptographisch geeignet ist, sei x_0 die x-Koordinate eines Kurvenpunkts P, der eine Untergruppe mit großer primer Ordnung d erzeugt und sei $Q_A = x(p_A*P)$ der öffentliche Schlüssel des Benutzers des ersten Rechners.

[0343] Damit der zweite Rechner eine Signatur (r, s) einer Nachricht m mit dem Hash-Wert $0 < h(m) < d$ verifizieren kann, sind folgende, in dem Ablaufdiagramm **1200** in Fig. 12 dargestellten Verfahrensschritte vorgesehen.

[0344] Zunächst wird das Zertifikat des öffentlichen Schlüssels Q_A des Benutzers des ersten Rechners überprüft und bei Ungültigkeit des Zertifikats wird von dem zweiten Rechner eine Fehlermeldung ausgegeben (Schritt **1201**).

[0345] Ferner wird geprüft, ob der erste Wert der Signatur r Element des Körpers $GF(2^m)$ ist und ob der erste Wert r ungleich 0 ist sowie ob der zweite Wert der Signatur s größer als 0 und kleiner als d ist. Ist eine der Bedingungen nicht erfüllt, so wird von dem zweiten Rechner eine Fehlermeldung generiert und ausgegeben (Schritt **1202**).

[0346] Ein vierter Zwischenwert x wird in einem weiteren Schritt gebildet mittels Multiplikation des ersten Werts r der Signatur mit dem öffentlichen Schlüssel Q_A des Benutzers des ersten Rechners (Schritt **1203**).

[0347] Ferner wird ein fünfter Zwischenwert y gebildet mittels Multiplikation des zweiten Werts s der Signatur (r, s) mit dem ersten Wert r der Signatur (Schritt **1204**).

[0348] Ein sechster Zwischenwert z wird gebildet mittels Multiplikation des Hash-Wertes h(m) mit der x-Koordinate x_0 des Kurvenpunkts P (Schritt **1205**).

[0349] Die EC-DSA-Signatur wird als gültig akzeptiert, wenn das quadratische Polynom q über dem endlichen Körper GF(2^m) über den vierten Zwischenwert x, den fünften Zwischenwert y sowie den sechsten Zwischenwert z gleich 0 ist (Schritt **1206**).

[0350] Es ist anzumerken, dass die Verfahrensschritte **1203**, **1204** und **1205** Skalar-Multiplikationen mit Punkten der elliptischen Kurve sind. Zur Berechnung des Werts des quadratischen Polynoms q(x, y, z) ist lediglich die Arithmetik des Grundkörpers erforderlich.

[0351] Das Polynom $q(x_1, x_2, x_3)$ hat die Form:

$$q(x_1, x_2, x_3) = x_3^2(x_1 + x_2)^2 + x_3x_1x_2 + x_1^2x_2^2 + a_6, \quad (46)$$

wobei a_6 der entsprechende Parameter der elliptischen Kurve E ist.

[0352] Um EC-GDSA-Signaturen erzeugen zu können, sind die Arithmetik des Grundkörpers, ein kryptographisch guter Zufallszahlen-Generator und eine Arithmetik zum Rechnen in Z/dZ, wobei d die Ordnung der Punktegruppe der elliptischen Kurve E ist, vorgesehen.

[0353] Bei der im Weiteren beschriebenen Protokollvariante werden von den berechneten Punkten der elliptischen Kurve wiederum nur die x-Koordinaten verwendet.

[0354] Sei im folgendem E eine elliptische Kurve, so dass die Punktegruppe kryptographisch geeignet ist, sei x_0 die x-Koordinate eines Kurvenpunktes P, der eine Untergruppe mit großer primer Ordnung d erzeugt und sei p_A der geheime Schlüssel und $Q_A = x(p_A^{-1}*P)$ der öffentliche Schlüssel des ersten Rechners.

[0355] Damit der erste Rechner den Hash-Wert $0 < h(m) < d$ einer Nachricht m signieren kann, sind folgende, in einem Ablaufdiagramm **1300** in Fig. 13 dargestellten Verfahrensschritte vorgesehen, welche von dem ersten Rechner durchgeführt werden.

[0356] In einem ersten Schritt (Schritt **1301**) wird eine Zufallszahl z erzeugt, wobei gilt: $0 < z < d$.

[0357] In einem weiteren Schritt (Schritt **1302**) wird ein erster Zwischenwert r gebildet mittels skalarer Multiplikation der Zufallszahl z mit der x-Koordinate x_0 des Kurvenpunktes P.

[0358] Ein zweiter Zwischenwert s wird gebildet mittels Multiplikation modulo d des geheimen Schlüssels p_A des Benutzers des ersten Rechners mit der Differenz des Produkts der Zufallszahl z mit dem ersten Zwischenwert r und dem Hash-Wert h(m) (Schritt **1303**).

[0359] Ist der zweite Zwischenwert s gleich dem Wert 0, so wird das Verfahren mit Schritt **1301** erneut durchgeführt.

[0360] Ist der zweite Zwischenwert s ungleich 0, so werden der erste Zwischenwert r und der zweite Zwischenwert s als Zahlentupel ausgegeben, wobei das Zahlentupel (r, s) eine gültige Signatur des Hash-Werts h(m) der Nachricht m darstellt.

[0361] Es ist anzumerken, dass die Schritte **1302** und **1303** in dem Körper Z/dZ berechnet werden.

[0362] Bei dem Verfahren zur Verifikation von EC-GDSA-Signaturen werden erfindungsgemäß wiederum lediglich die x-Koordinaten der Punkte auf den elliptischen Kurven verwendet. Um das Protokoll zur Verifikation von EC-GDSA-Signaturen zu implementieren ist lediglich die Arithmetik des Grundkörpers erforderlich.

[0363] Sei im Folgenden E eine elliptische Kurve mit der Kurvengleichung

$$y^2 + x \cdot y = x^3 + a_2 \cdot x^2 + a_6, \quad (47)$$

so dass die Punktegruppe der elliptischen Kurve kryptographisch geeignet ist, sei x_0 die x-Koordinate eines Kurvenpunktes P, der eine Untergruppe mit großer primer Ordnung d erzeugt und sei $Q_A = x(p_A^{-1} * P)$ der öffentliche Schlüssel des Benutzers des ersten Rechners.

[0364] Damit der zweite Rechner eine Signatur (r, s) eine Nachricht m mit dem Hashwert $0 < h(m) < d$ verifizieren kann, werden folgende Verfahrensschritte, welche in einem Ablaufdiagramm **1400** in Fig. 14 dargestellt sind, von dem zweiten Rechner durchgeführt.

[0365] Zunächst wird in einem ersten Schritt (Schritt **1401**) das Zertifikat des öffentlichen Schlüssels Q_A des Benutzers des ersten Rechners überprüft und bei Ungültigkeit des Zertifikats wird von dem zweiten Rechner eine Fehlermeldung ausgegeben.

[0366] Ferner wird geprüft, ob der erste Zwischenwert r der Signatur Element des Körpers $GF(2^m)$ ist, sowie ob der erste Zwischenwert r ungleich 0 ist und ob der zweite Zwischenwert s größer als 0 und kleiner als d ist.

[0367] Ist eine der Bedingungen nicht erfüllt, wird ein Fehlersignal ausgegeben (Schritt **1402**).

[0368] In einem weiteren Schritt (Schritt **1403**) wird ein vierter Zwischenwert x gebildet mittels Multiplikation des öffentlichen Schlüssels Q_A des Benutzers des ersten Rechners mit dem zweiten Zwischenwert s.

[0369] Ferner wird ein fünfter Zwischenwert y gebildet mittels Multiplikation des ersten Zwischenwertes r mit sich selbst (Schritt **1404**).

[0370] Ein sechster Zwischenwert z wird gebildet mittels Skalar-Multiplikation des Hash-Werts h(m) mit der x-Koordinate x_0 des Punktes P auf der elliptischen Kurve (Schritt **1405**).

[0371] Anschließend wird ein quadratisches Polynom q über dem endlichen Körper $GF(2^m)$ über den ersten Zwischenwert x, den zweiten Zwischenwert y und den dritten Zwischenwert z ermittelt und es wird überprüft, ob das quadratische Polynom gleich 0 ist.

[0372] Ist dies der Fall, dann wird die digitale Signatur als gültig angesehen (Schritt **1406**).

[0373] Das heißt, dass zur Berechnung des Wertes q(x, y, z) lediglich die Arithmetik des Grundkörpers notwendig ist.

[0374] Das quadratische Polynom $q(x_1, x_2, x_3)$ hat die Form:

$$q(x_1, x_2, x_3) = x_3^2(x_1 + x_2)^2 + x_3 x_1 x_2 + x_1^2 x_2^2 + a_6, \quad (48)$$

wobei a_6 der entsprechende Parameter der elliptischen Kurve E ist.

[0375] Es ist anzumerken, dass die Verfahrensschritte **1403**, **1404**, **1405** Skalar-Multiplikationen mit Punkten der elliptischen Kurve sind.

[0376] Bei der Implementierung der Signaturüberprüfung müssen die Ergebnispunkte der drei Skalarmultiplikationen nicht in die affine Koordinatendarstellung transformiert werden.

[0377] Das Polynom q(x, y, z) zum Testen des Ergebnisses kann auf einfache Weise für Punkte in projektiver Darstellung verallgemeinert werden.

[0378] Auf diese Weise ist das oben beschriebene Verfahren zur Verifikation effizienter und das Polynom kann auch auf die Fälle verallgemeinert werden, bei denen eines oder mehrere Ergebnisse der Skalar-Multiplikationen der unendlich ferne Punkt O ist.

[0379] Bei allen Verfahren zur Signaturüberprüfung kann auch eine Variante verwendet werden, bei der durch den Wert s dividiert wird und auf diese Weise lediglich zwei Skalar-Multiplikationen anfallen.

[0380] Im Weiteren wird beschrieben, wie die oben beschriebene Verfahrensüberprüfung elektronischer Unterschriften mittels Verfahren, bei denen vollständig auf die Berechnung und Verwendung von y-Koordinaten verzichtet wird, sowohl auf ähnliche Körper anderer Charakteristiken als auch auf projektive Koordinatendarstellungen von Kurvenpunkten verallgemeinert werden kann.

[0381] Dabei wird lediglich folgender Teilaspekt beschrieben:
Bei allen im vorliegenden Dokument betrachteten Verfahren zur Erzeugung elektronischer Signaturen ist es wahrend der Überprüfung einer Signatur vorgesehen, zu testen, ob drei berechnete Punkte P, Q und R gemäß der Gruppenverknüpfung der elliptischen Kurve in der Beziehung P + Q = R stehen.

[0382] Sind von P, Q und R nur die x-Koordinaten bekannt, kann dieser Test nicht mehr durchgeführt werden. Es ist aber möglich, zu testen, ob die Punkte einer einfachen Polynom-Gleichung genügen. Dabei liefert das Polynom genau dann den Wert 0, wenn ±p ±Q ±R = O gilt (in der affinen Version unter der Voraussetzung, dass alle drei Punkte verschieden von dem unendlich fernen Punkt O sind). Der auftretende Sicherheitsverlust stellt für die praktische Anwendung der Signaturverfahren keine Einschränkung dar.

[0383] Im Folgenden werden die entsprechenden Polynome und Sonderfälle in Abhängigkeit von der Charakteristik des endlichen Körpers, über dem die elliptische Kurve definiert ist, beschrieben.

[0384] Für elliptische Kurvenen über endlichen Körpern der Charakteristik 2 lautet die allgemeine Kurvengleichung:

$$y^2 + x \cdot y = x^3 + a_2 x^2 + a_6, \tag{49}$$

wobei a_6 die Diskriminate ist und es gilt daher für nicht-singuläre Kurven a_6 ungleich 0.

[0385] Aus der Additionsformel ergibt sich, dass für die Punkte $P = (x_1, y_1)$, $Q = (x_2, y_2)$, $R = (x_3, y_3)$ mit P + Q = R und P, Q, R ≠ O das Polynom:

$$q(x_1, x_2, x_3) = x_3^2(x_1 + x_2)^2 + x_3 x_1 x_2 + x_1^2 x_2^2 + a_6 \tag{50}$$

den Wert 0 hat.

[0386] Gilt P = Q und $x_1 = x_2 \neq 0$, so folgt aus der Formel zur Verdoppelung von Punkten die folgende Vorschrift:

$$x_3 \cdot x_1^2 + x_1^4 + a_6 = 0, \tag{51}$$

die in Gleichung (50) als Spezialfall bereits enthalten ist.

[0387] Gilt P = Q und 2P = O (d.h. $x_1 = x_2 = 0$), dann folgt aus Gleichung (51), dass $a_6 = 0$ ist. Dies ist aber nicht möglich, weil vorausgesetzt wurde, dass die elliptische Kurve nichtsingulär ist.

[0388] Für elliptische Kurven über endlichen Körpern der Charakteristik 3 lautet die allgemeine Kurvengleichung:

$$y^2 = x^3 + a_2 \cdot x^2 + a_6, \tag{52}$$

wobei $-a_2^3 \cdot a_6$ die Diskriminate ist und es gilt daher für nicht-singuläre elliptische Kurven $a_2^3 \cdot a_6 \neq 0$.

[0389] Aus der Additionsformel ergibt sich, dass für Punkte $P = (x_1, y_1)$, $Q = (x_2, y_2)$ und $R = (x_3, y_3)$ mit P + Q = R und P, Q, R ≠ O das Polynom

$$q(x_1, x_2, x_3) = x_3^2(x_1 - x_2)^2 + x_3(x_1 x_2(x_1 + x_2 - a_2) - a_6) + \\ + x_1^2 x_2^2 - a_6(x_1 + x_2 + a_2) \tag{53}$$

den Wert 0 hat.

[0390] Gilt P = Q und $y_1 = y_2 \neq 0$, so folgt aus der Formel zur Verdoppelung von Punkten die Gleichung

$$x_3(x_1^3 + a_2x_1^2 + a_6) = x_1^4 + a_6(x_1 - a_2), \quad (54)$$

die in Vorschrift (55) als Spezialfall bereits enthalten ist.

[0391] Gilt P = Q und 2P = O (d.h. $y_1 = y_2 = 0$), dann folgt aus der Resultante von Gleichung (54) und der Kurvengleichung die Bedingung $a_3^2a_6 = 0$. Dies ist aber nicht möglich, weil vorausgesetzt wurde, dass die elliptische Kurve nicht-singulär ist.

[0392] Für elliptische Kurven über endlichen Körpern der Charakteristik größer als 3 lautet die allgemeine Kurvengleichung:

$$y^2 = x^3 + a_4x + a_6, \quad (55)$$

wobei $-16(4a_4^3 + 27a_6^2)$ die Diskriminate ist und es gilt daher für nicht-singuläre Kurven $4a_4^3 + 27a_6^2 \neq 0$.

[0393] Aus der Additionsformel ergibt sich, dass für Punkte $P = (x_1, y_1)$, $Q = (x_2, y_2)$ und $R = (x_3, y_3)$ mit P + Q = R und P, Q, R ungleich 0 das Polynom

$$q(x_1, x_2, x_3) = x_3^2(x_1 - x_2)^2 + 2x_3((a_4 + x_1x_2)(x_1 + x_2) + 2a_6) + \\ +(x_1x_2 - a_4)^2 - 4a_6(x_1 + x_2) \quad (56)$$

den Wert 0 hat.

[0394] Gilt P = Q und $y_1 = y_2 \neq 0$, so folgt aus der Formel zur Verdoppelung von Punkten die Gleichung

$$-4x_3(x_1^3 + a_4x_1 + a_6) + (x_1^2 - a_4)^2 - 8a_6x_1 = 0, \quad (57)$$

die in Vorschrift (56) als Spezialfall bereits enthalten ist.

[0395] Gilt P = Q und 2P = O (d.h. $y_1 = y_2 = 0$), dann folgt aus der Resultanten von Vorschrift (57) und der Kurvengleichung die Bedingung $4a_4^3 + 27a_6^2 = 0$. Dies ist aber nicht möglich, weil vorausgesetzt wurde, dass die Kurve nicht-singulär ist.

[0396] Werden von den Polynomen (50, 53, 56) entsprechende Polynome für projektive Koordinatendarstellungen verwendet, können die Inversionen im Grundkörper am Ende der Skalar-Multiplikationen eingespart werden, was einen weiteren Performance-Vorteil bei der Signaturüberprüfung mit sich bringt.

[0397] Außerdem bleiben die entsprechenden Polynome für projektive Koordinatendarstellungen auch für Sonderfälle wichtig, bei denen einer oder mehrere der Punkte P, Q und R der unendlich ferne Punkt O ist.

[0398] Dieses wird im Folgenden näher erläutert.

[0399] Seien x_i/z_i die projektiv dargestellten x-Koordinaten der Punkte P_i für $1 \leq i \leq 3$. Der unendlich ferne Punkt O besitzt dabei die projektive Darstellung (x, z) mit $x \neq 0$ und z = 0.

[0400] Für elliptische Kurven über endlichen Körpern der Charakteristik 2 lautet die projektive Form von (50):

$$x_3^2(x_1z_2 + x_2z_1)^2 + x_3x_1x_2z_3z_1z_2 + x_1^2x_2^2x_3^2 + a_6z_1^2z_2^2z_3^2 = 0. \quad (58)$$

[0401] Für den Fall $P_3 = O$, d.h. $z_3 = 0$, $x_3 \neq 0$ ergibt sich aus Vorschrift (58):

$$x_3^2(x_1z_2 + x_2z_1)^2 = 0. \quad (59)$$

[0402] Das ist genau dann der Fall, wenn $x(P_1) = x(P_2)$ oder $P_1 = P_2 = O$ gilt. Damit wird dieser Fall korrekt behandelt.

[0403] Der Fall $P_1 = O$ ist symmetrisch zum vorhergehenden Fall.

[0404] Der Fall $P_2 = O$ ist ebenfalls symmetrisch zum vorhergehenden Fall.

[0405] Für elliptische Kurven über endlichen Körpern der Charakteristik 3 lautet die projektive Form von (53):

$$x_3^2(x_1z_2 - x_2z_1)^2 + x_3z_3(x_1x_2(x_1x_2 + x_2z_1 - a_2z_1z_2) - a_6z_1^2z_2^2) + \\ + z_3^2(x_1^2x_2^2 - a_6z_1z_2(a_2z_1z_2 + x_1z_2 + x_2z_1)) = 0 \quad (60)$$

[0406] Für den Fall $P_3 = O$, d.h. $z_3 = 0$, $x_3 \neq 0$ ergibt sich aus Vorschrift (60):

$$x_3^2(x_1z_2 - x_2z_1)^2 = 0. \quad (61)$$

[0407] Das ist genau dann der Fall, wenn $x(P_1)$, $x(P_2)$ oder $P_1 = P_2 = O$ gilt. Damit wird dieser Fall korrekt behandelt.

[0408] Der Fall $P_1 = O$ ist symmetrisch zum vorhergehenden Fall.

[0409] Der Fall $P_2 = O$ ist ebenfalls symmetrisch zum vorhergehenden Fall.

[0410] Für elliptische Kurven über endlichen Körpern der Charakteristik größer als 3 lautet die projektive Form von Vorschrift (56):

$$x_3^2(x_1z_2 - x_2Z_1)^2 + 2x_3z_3((a_4z_1z_2 + x_1x_2)(x_1z_2 + x_2z_1) + 2a_6z_1^2z_2^2) + \\ + z_3^2(x_1x_2 - a_4z_1z_2)^2 - 4a_6z_1z_2z_3^2(x_1z_2 + x_2z_1) = 0 \quad (62)$$

[0411] Für den Fall $P_3 = O$, d.h. $z_3 \neq 0$, $x_3 \neq 0$ ergibt sich gemäß Vorschrift (62):

$$x_3^2(x_1z_2 - x_2z_1)^2 = 0. \quad (63)$$

[0412] Das ist genau dann der Fall, wenn $x(P_1) = x(P_2)$ oder $P_1 = P_2 = O$ gilt. Damit wird dieser Fall korrekt behandelt.

[0413] Der Fall $P_1 = O$ ist symmetrisch zum vorhergehenden Fall.

[0414] Der Fall $P_2 = O$ ist ebenfalls symmetrisch zum vorhergehenden Fall.

[0415] In diesem Dokument sind folgende Veröffentlichungen zitiert:

[1] T.S. Messerges et al, Power analysis attacks of modular exponentiation in smartcards, Proceedings of CHES'99, LNCS 1717, Springer, Seiten 144-154, 1999

[2] P. Kocher et al, Differential power analysis, Proceedings of CRYPTO'99, LNCS 1666, Springer, Seiten 388-397, 1999

[3] I. Biehl et al, Differential fault attacks on elliptic curve cryptosystems, Proceedings of CRYPTO'00, LNCS 1880, Springer, Seiten 131-146, 2000

[4] D. Boneh et al, On the importance of checking cryptographic protocols for faults, Proceedings of EUROCRYPT'97, LNCS 1233, Springer, Seiten 37-51, 1997

[5] P. Kocher, Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems, Proceedings of CRYPTO'96, LNCS 1109, Springer Verlag, Seiten 104-113, 1996

[6] J.-S. Coron, Resistance against differential power analysis for elliptic curve cryptosystems, Proceedings of CHES'99, LNCS 1717, Springer Verlag, Seiten 292-302, 1999

[7] P.L. Montgomery, Speeding up the pollard and elliptic curve methods of factorization, mathematics of computation, Vol. 48, Nr. 177, Seiten 243-264, 1987

[8] A.J. Menezes, Elliptic curve public-key cryptosystems, Kluwer Academic Publishers, Seiten 209-224, 1993

[9] K. Ohgishi et al, Elliptic curve signature scheme with no y coordinate, The 1999 Symposium on cryptography and information security, SCIS'99, The Institute of Electronics, Information and Communication Engineers, 1999

[10] Izu Tetsuya, Elliptic curve exponentiation without y-coordinate, Technical Report IT-98-33, ISEC 98-86, SST 98-129, IEICE The Institute of Electronics, Information and Communication Engineers, März 1999

Patentansprüche

1. Verfahren zum rechnergestützten Erzeugen und Verifizieren einer digitalen Signatur,
– bei dem von einem ersten Rechner folgende Verfahrensschritte durchgeführt werden:
– eine elliptische Kurve über einem Körper der Charakteristik 2 wird bereitgestellt,

– mindestens ein Punkt, der auf der elliptischen Kurve liegt, wird ausgewählt oder ermittelt,
– nur eine Koordinate des Punktes wird gespeichert,
– Daten werden gemäß einem vorgegebenen Verfahren zum Erzeugen einer digitalen Signatur kryptographisch bearbeitet, wobei bei dem gesamten kryptographischen Verfahren nur die eine Koordinate des Punktes verwendet wird,
– bei dem die kryptographisch bearbeiteten Daten von dem ersten Rechner zu einem zweiten Rechner übertragen werden, wobei von auf der elliptischen Kurve sich befindenden Punkten, welche von dem ersten Rechner zu dem zweiten Rechner übertragen werden, jeweils nur die eine Koordinate des jeweiligen Punktes übertragen wird, und – bei dem von dem zweiten Rechner zumindest folgende Verfahrensschritte durchgeführt werden:
– die übertragenen kryptographisch bearbeiteten Daten werden gemäß einem Verfahren zum Verifizieren einer digitalen Signatur zusätzlich bearbeitet, wobei bei der gesamten Verifizierung der digitalen Signatur nur die eine Koordinate eines jeweiligen Punktes auf der elliptischen Kurve verwendet wird, wobei überprüft wird, ob die jeweils einen Koordinaten dreier Punkte auf der elliptischen Kurve eine Polynomgleichung erfüllen.

2. Verfahren nach Anspruch 1, bei dem als die eine Koordinate des jeweiligen Punktes auf der elliptischen Kurve die x-Koordinate des Punktes verwendet wird.

3. Verfahren nach Anspruch 1, bei dem zum Erzeugen einer digitalen Signatur und zum Verifizieren einer digitalen Signatur jeweils eines der folgenden Verfahren eingesetzt wird:
– ein Verfahren zum Erzeugen einer digitalen Signatur und ein Verfahren zum Prüfen einer digitalen Signatur, welches auf einem Verfahren gemäß ElGamal basiert,
– ein Verfahren zum Erzeugen einer digitalen Signatur und ein Verfahren zum Prüfen einer digitalen Signatur, welches auf einem Verfahren gemaß EC-DSA basiert,
– ein Verfahren zum Erzeugen einer digitalen Signatur und ein Verfahren zum Prüfen einer digitalen Signatur, welches auf einem Verfahren gemäß EC-GDSA basiert.

4. System zum Erzeugen und Verifizieren einer digitalen Signatur mit einem ersten Rechner und einem zweiten Rechner, wobei der erste Rechner und der zweite Rechner über ein Telekommunikationsnetz miteinander gekoppelt sind,
– bei dem der erste Rechner eine Prozessoreinheit aufweist, die derart eingerichtet ist, dass folgende Verfahrensschritte durchführbar sind:
– es wird eine elliptische Kurve über einem Körper der Charakteristik 2 bereitgestellt,
– es wird mindestens ein Punkt, der auf der elliptischen Kurve liegt, ausgewählt oder ermittelt,
– es wird nur eine Koordinate des Punktes auf der elliptischen Kurve gespeichert,
– Daten werden gemäß einem vorgegebenen Verfahren zum Erzeugen einer digitalen Signatur kryptographisch bearbeitet, wobei bei dem gesamten kryptographischen Verfahren nur die eine Koordinate des Punktes verwendet wird,
– die kryptographisch bearbeiteten Daten werden von dem ersten Rechner an den zweiten Rechner über das Telekommunikationsnetz gesendet, wobei für auf der elliptischen Kurve sich befindende Punkte jeweils nur die eine Koordinate des jeweiligen Punktes übertragen wird,
– bei dem der zweite Rechner eine Prozessoreinheit aufweist, die derart eingerichtet ist, dass folgende Verfahrensschritte durchführbar sind:
– die von dem ersten Rechner gesendeten kryptographisch bearbeiteten Daten werden empfangen,
– die empfangenen kryptographisch bearbeiteten Daten werden gemäß einem Verfahren zum Verifizieren einer digitalen Signatur zusätzlich bearbeitet, wobei bei der gesamten Verifizierung der digitalen Signatur nur die eine Koordinate des jeweiligen Punktes auf der elliptischen Kurve verwendet wird, wobei überprüft wird, ob die jeweils einen Koordinaten dreier Punkte auf der elliptischen Kurve eine Polynomgleichung erfüllen.

5. Verfahren zum rechnergestützten Verifizieren einer digitalen Signatur,
– bei dem kryptographisch bearbeitete Daten von einem Rechner empfangen werden, wobei von auf einer elliptischen Kurve über einem Körper der Charakteristik 2 sich befindenden Punkte, welche empfangen werden, jeweils nur die eine Koordinate des jeweiligen Punktes empfangen wird, und
– bei dem von dem Rechner zumindest folgende Verfahrensschritte durchgeführt werden:
– die empfangenen kryptographisch bearbeiteten Daten werden gemäß einem Verfahren zum Verifizieren einer digitalen Signatur bearbeitet, wobei bei der gesamten Verifizierung der digitalen Signatur nur die eine Koordinate eines jeweiligen Punktes auf der elliptischen Kurve verwendet wird, wobei überprüft wird, ob die jeweils einen Koordinaten dreier Punkte auf der elliptischen Kurve eine Polynomgleichung erfüllen.

6. System zum Verifizieren einer digitalen Signatur mit einem Rechner,

bei dem der Rechner eine Prozessoreinheit aufweist, die derart eingerichtet ist, dass folgende Verfahrensschritte durchführbar sind:

– kryptographisch bearbeitet Daten werden empfangen, wobei von auf einer elliptischen Kurve über einem Körper der Charakteristik 2 sich befindenden Punkten, welche empfangen werden, jeweils nur die eine Koordinate des jeweiligen Punktes empfangen wird

– die empfangenen kryptographisch bearbeiteten Daten werden gemäß einem Verfahren zum Verifizieren einer digitalen Signatur bearbeitet, wobei bei der gesamten Verifizierung der digitalen Signatur nur die eine Koordinate eines jeweiligen Punktes auf der elliptischen Kurve verwendet wird, wobei überprüft wird, ob die jeweils einen Koordinaten dreier Punkte auf der elliptischen Kurve eine Polynomgleichung erfüllen.

Es folgen 11 Blatt Zeichnungen

Anhängende Zeichnungen

FIG 1

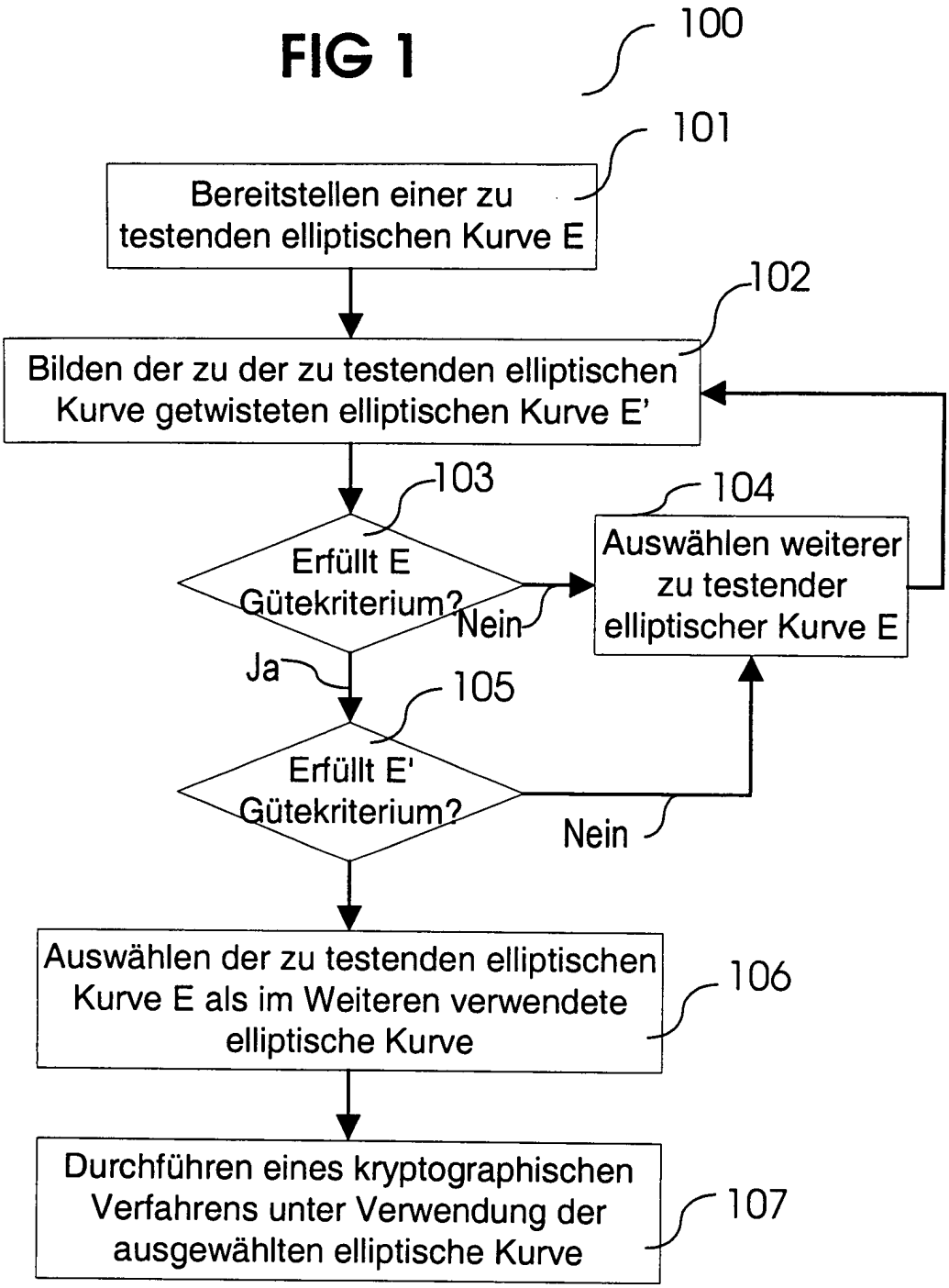


FIG 2

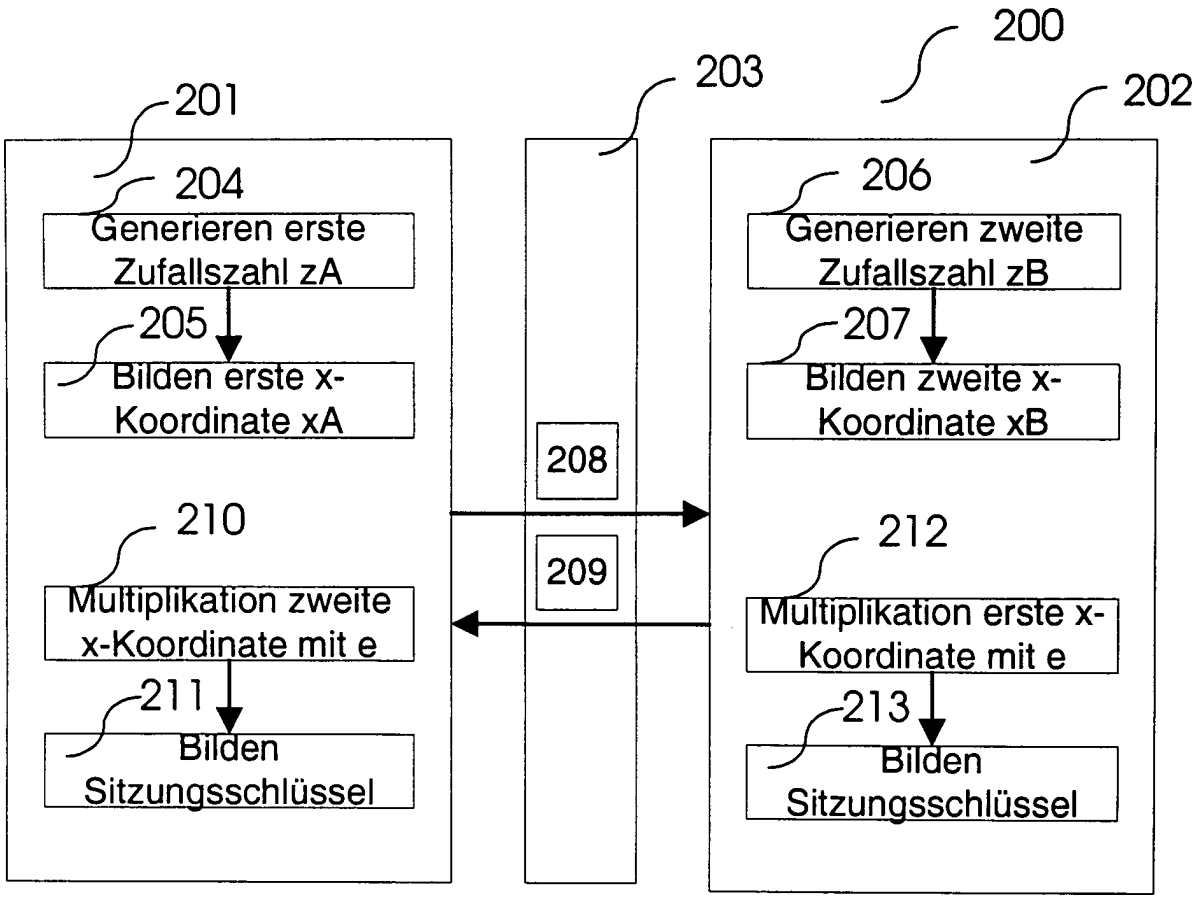

200
201
202
203
204
Generieren erste Zufallszahl zA
205
Bilden erste x-Koordinate xA
206
Generieren zweite Zufallszahl zB
207
Bilden zweite x-Koordinate xB
208
209
210
Multiplikation zweite x-Koordinate mit e
211
Bilden Sitzungsschlüssel
212
Multiplikation erste x-Koordinate mit e
213
Bilden Sitzungsschlüssel

FIG 3

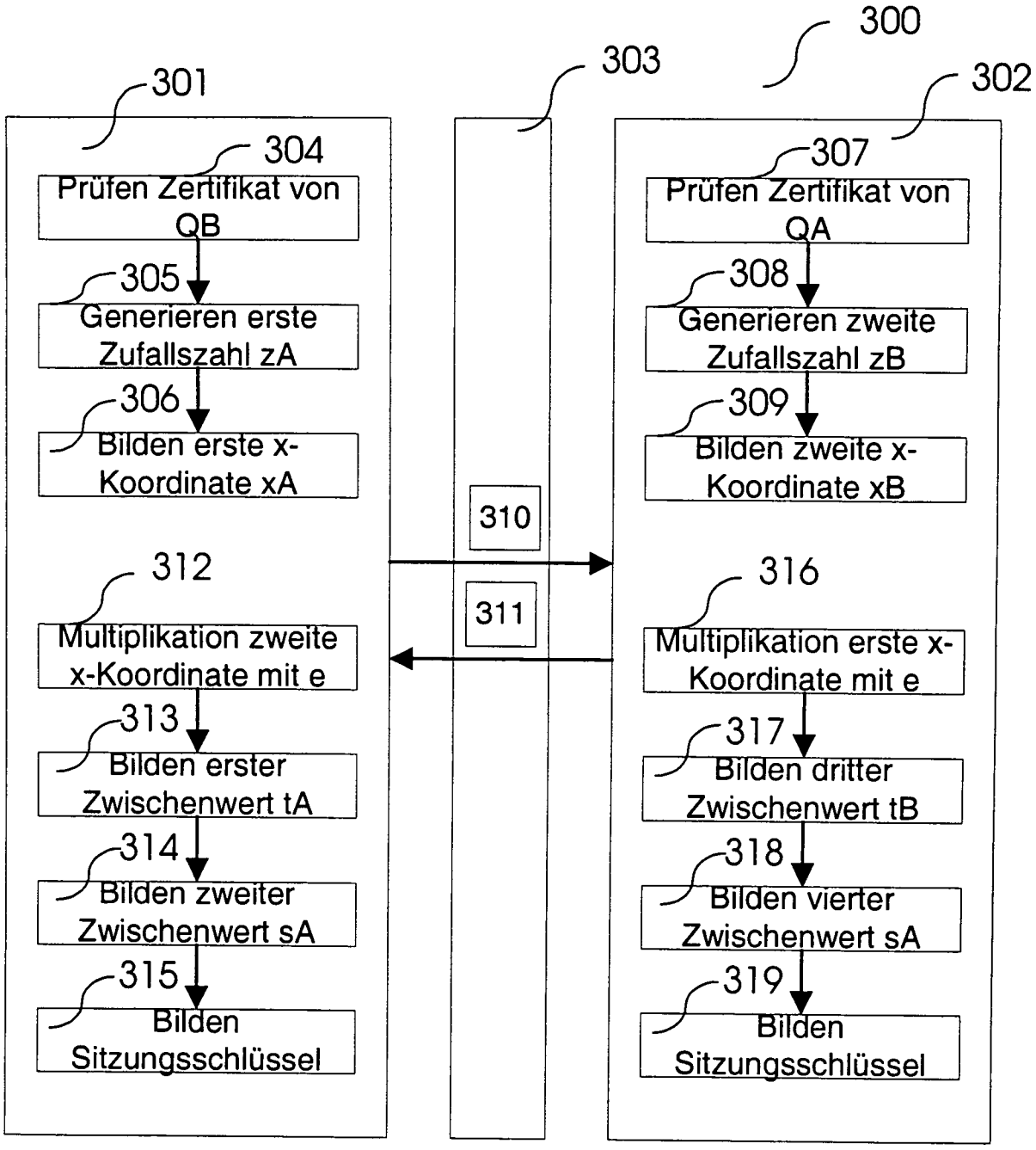

300
301
303
302
304
Prüfen Zertifikat von QB
305
Generieren erste Zufallszahl zA
306
Bilden erste x-Koordinate xA
307
Prüfen Zertifikat von QA
308
Generieren zweite Zufallszahl zB
309
Bilden zweite x-Koordinate xB
310
311
312
Multiplikation zweite x-Koordinate mit e
313
Bilden erster Zwischenwert tA
314
Bilden zweiter Zwischenwert sA
315
Bilden Sitzungsschlüssel
316
Multiplikation erste x-Koordinate mit e
317
Bilden dritter Zwischenwert tB
318
Bilden vierter Zwischenwert sA
319
Bilden Sitzungsschlüssel

FIG 4

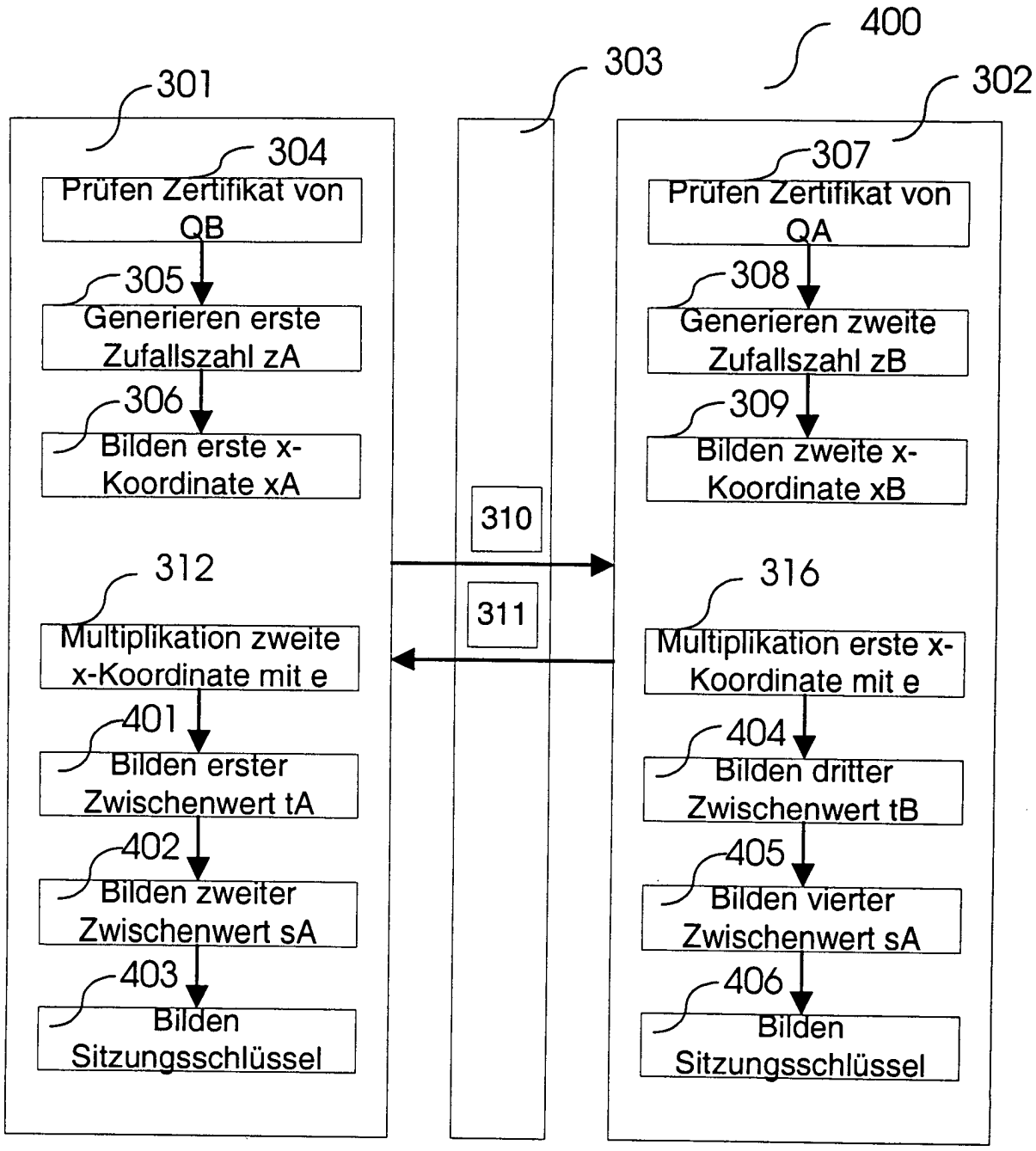

400
303
301
302
304
Prüfen Zertifikat von QB
307
Prüfen Zertifikat von QA
305
Generieren erste Zufallszahl zA
308
Generieren zweite Zufallszahl zB
306
Bilden erste x-Koordinate xA
309
Bilden zweite x-Koordinate xB
310
311
312
Multiplikation zweite x-Koordinate mit e
316
Multiplikation erste x-Koordinate mit e
401
Bilden erster Zwischenwert tA
404
Bilden dritter Zwischenwert tB
402
Bilden zweiter Zwischenwert sA
405
Bilden vierter Zwischenwert sA
403
Bilden Sitzungsschlüssel
406
Bilden Sitzungsschlüssel

FIG 5

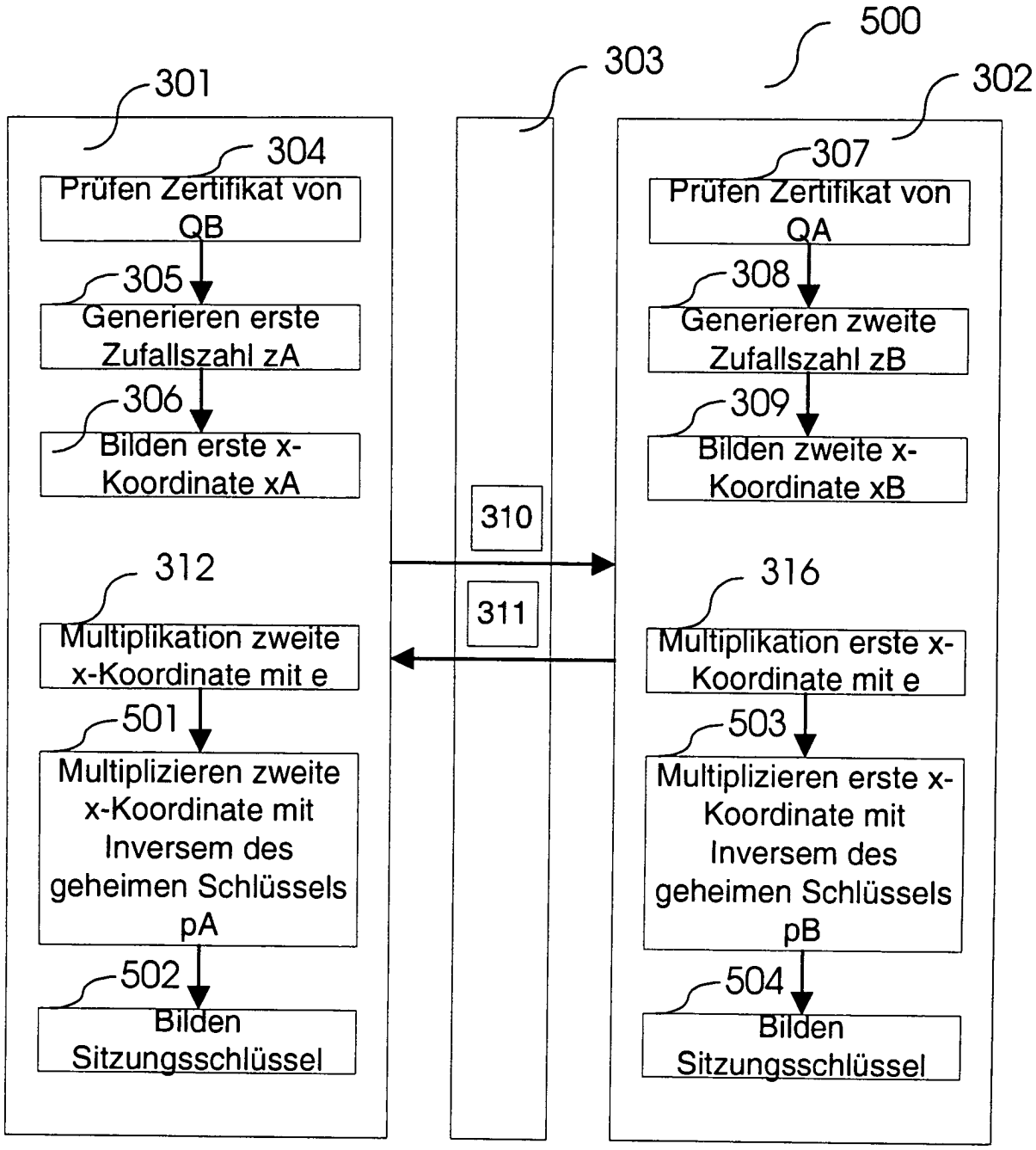

500
303
301
302
304
Prüfen Zertifikat von QB
307
Prüfen Zertifikat von QA
305
Generieren erste Zufallszahl zA
308
Generieren zweite Zufallszahl zB
306
Bilden erste x-Koordinate xA
309
Bilden zweite x-Koordinate xB
310
312
316
311
Multiplikation zweite x-Koordinate mit e
Multiplikation erste x-Koordinate mit e
501
503
Multiplizieren zweite x-Koordinate mit Inversem des geheimen Schlüssels pA
Multiplizieren erste x-Koordinate mit Inversem des geheimen Schlüssels pB
502
504
Bilden Sitzungsschlüssel
Bilden Sitzungsschlüssel

FIG 6

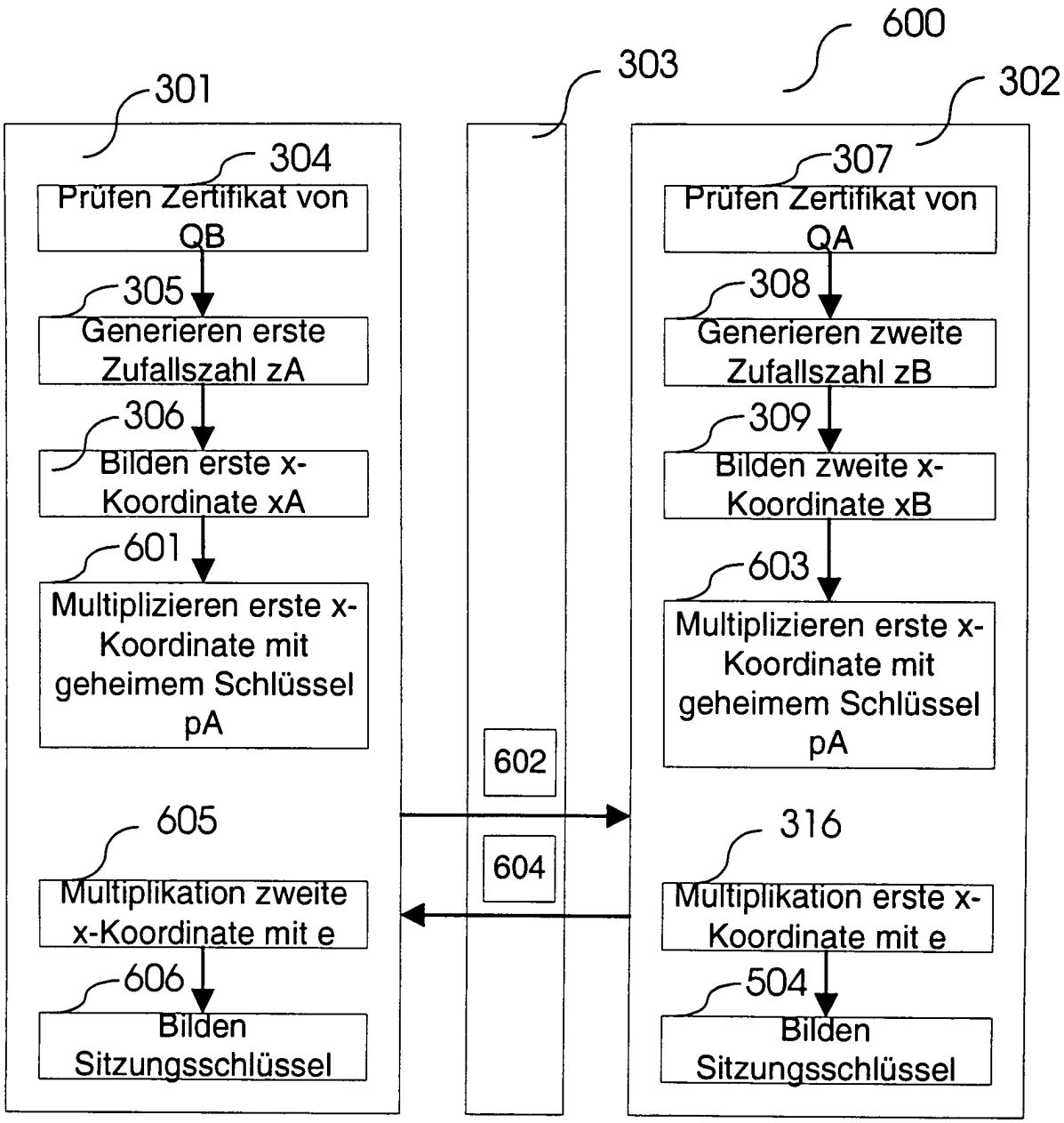

600
303
301
302
304
Prüfen Zertifikat von QB
307
Prüfen Zertifikat von QA
305
Generieren erste Zufallszahl zA
308
Generieren zweite Zufallszahl zB
306
Bilden erste x-Koordinate xA
309
Bilden zweite x-Koordinate xB
601
Multiplizieren erste x-Koordinate mit geheimem Schlüssel pA
603
Multiplizieren erste x-Koordinate mit geheimem Schlüssel pA
602
604
605
Multiplikation zweite x-Koordinate mit e
316
Multiplikation erste x-Koordinate mit e
606
Bilden Sitzungsschlüssel
504
Bilden Sitzungsschlüssel

FIG 7

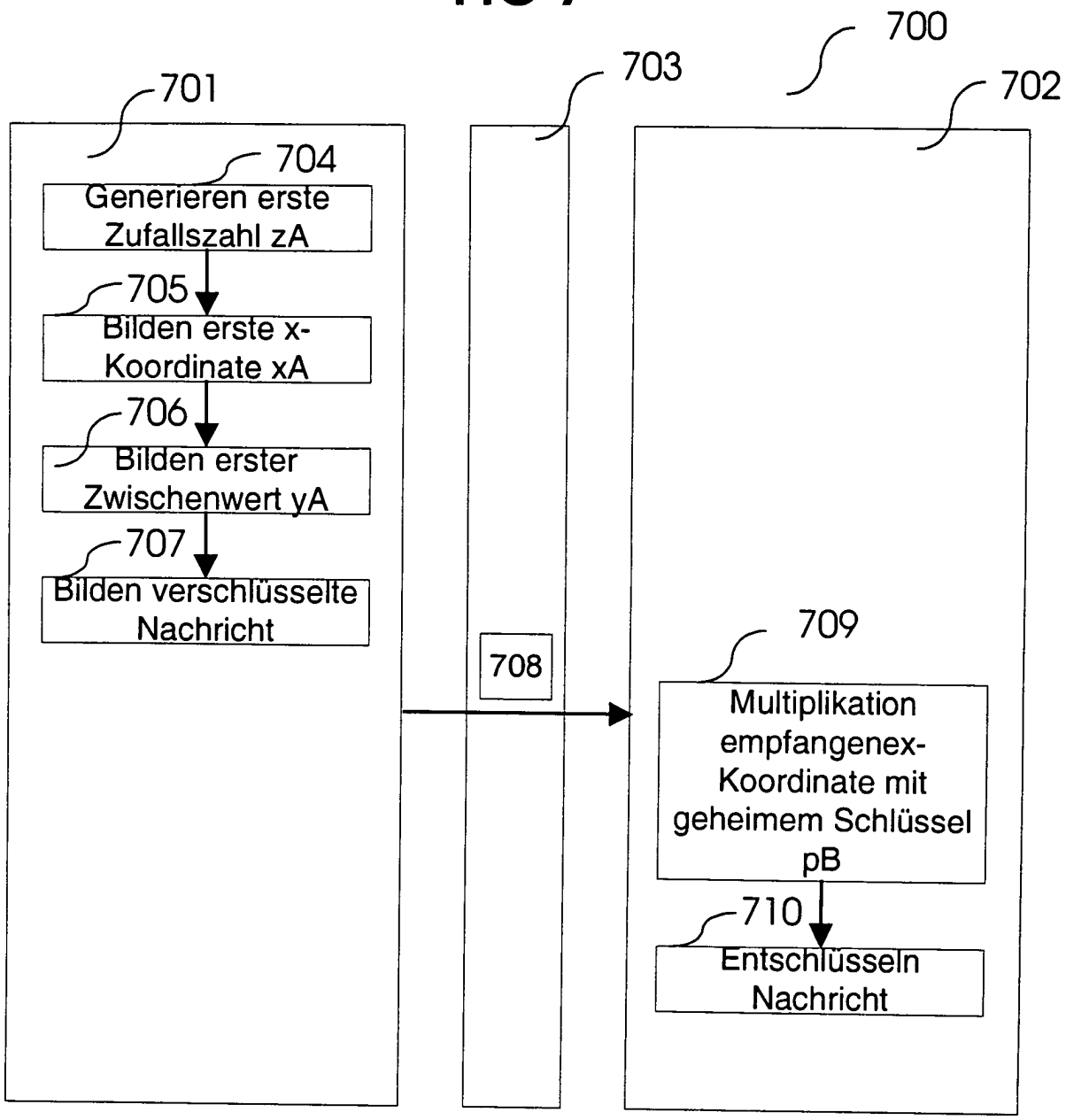

700
703
701
702
704
Generieren erste Zufallszahl zA
705
Bilden erste x-Koordinate xA
706
Bilden erster Zwischenwert yA
707
Bilden verschlüsselte Nachricht
708
709
Multiplikation empfangenex-Koordinate mit geheimem Schlüssel pB
710
Entschlüsseln Nachricht

FIG 8

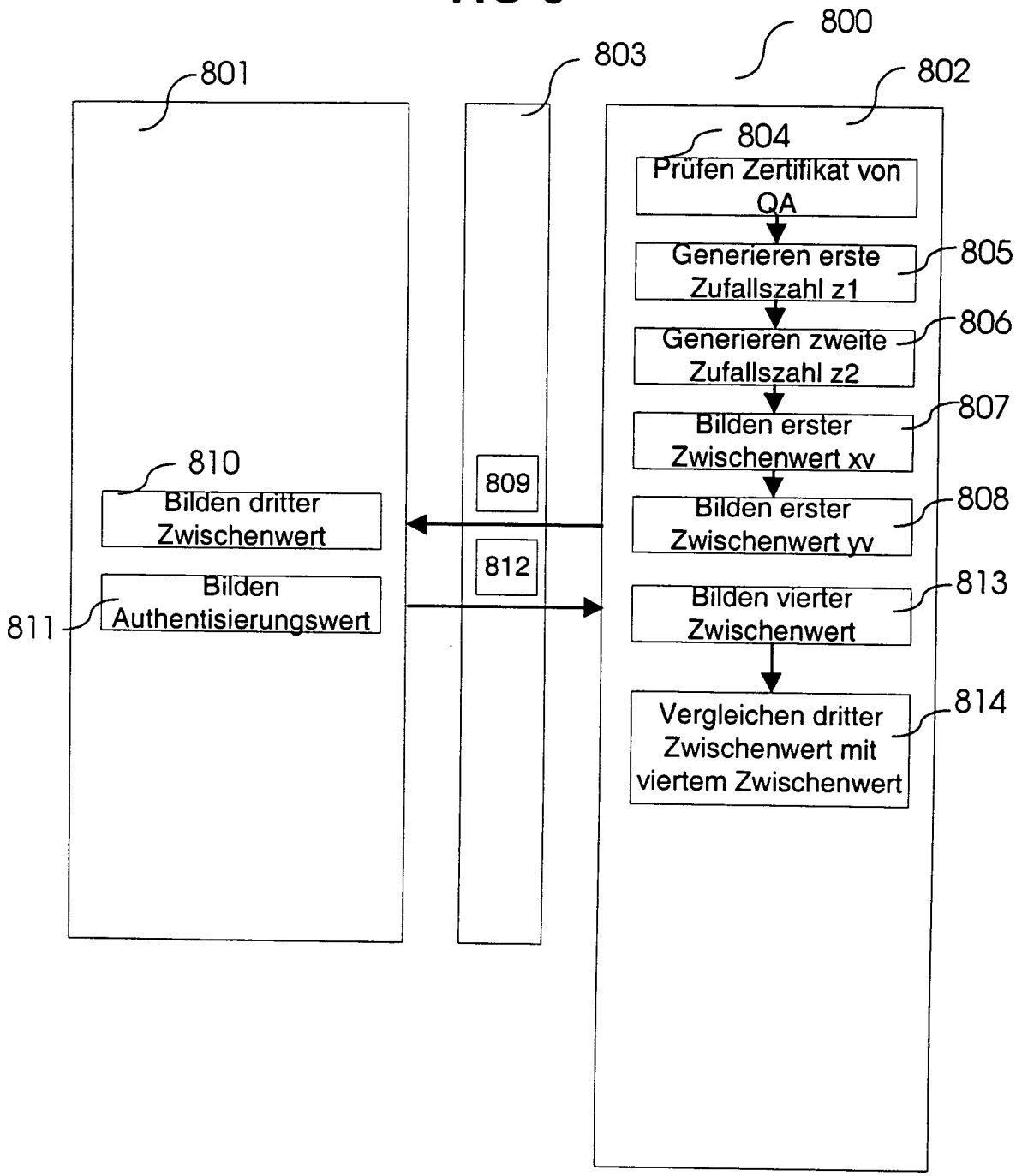

800
801
803
802
804
Prüfen Zertifikat von QA
Generieren erste Zufallszahl z1
805
Generieren zweite Zufallszahl z2
806
Bilden erster Zwischenwert xv
807
Bilden erster Zwischenwert yv
808
Bilden vierter Zwischenwert
813
Vergleichen dritter Zwischenwert mit viertem Zwischenwert
814
809
812
810
Bilden dritter Zwischenwert
811
Bilden Authentisierungswert

FIG 9

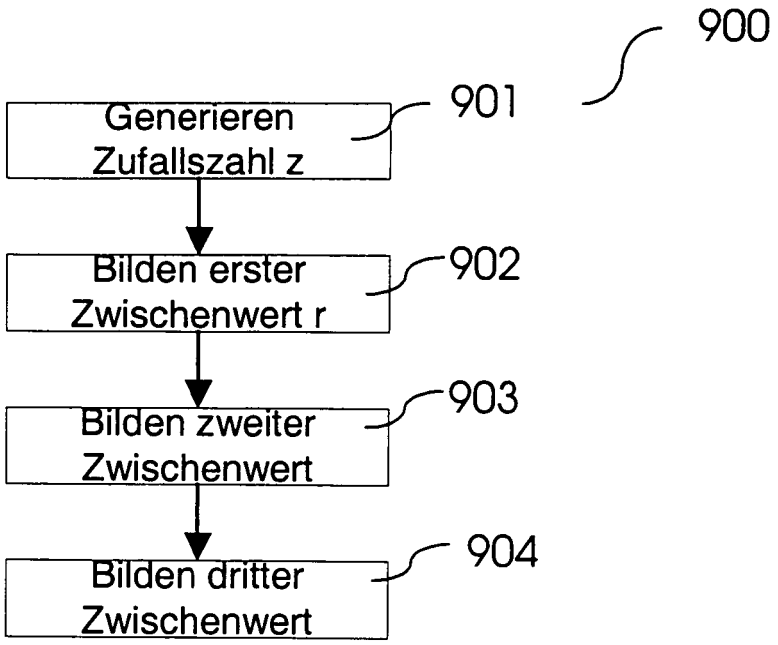

900
Generieren Zufallszahl z
901
Bilden erster Zwischenwert r
902
Bilden zweiter Zwischenwert
903
Bilden dritter Zwischenwert
904

FIG 10

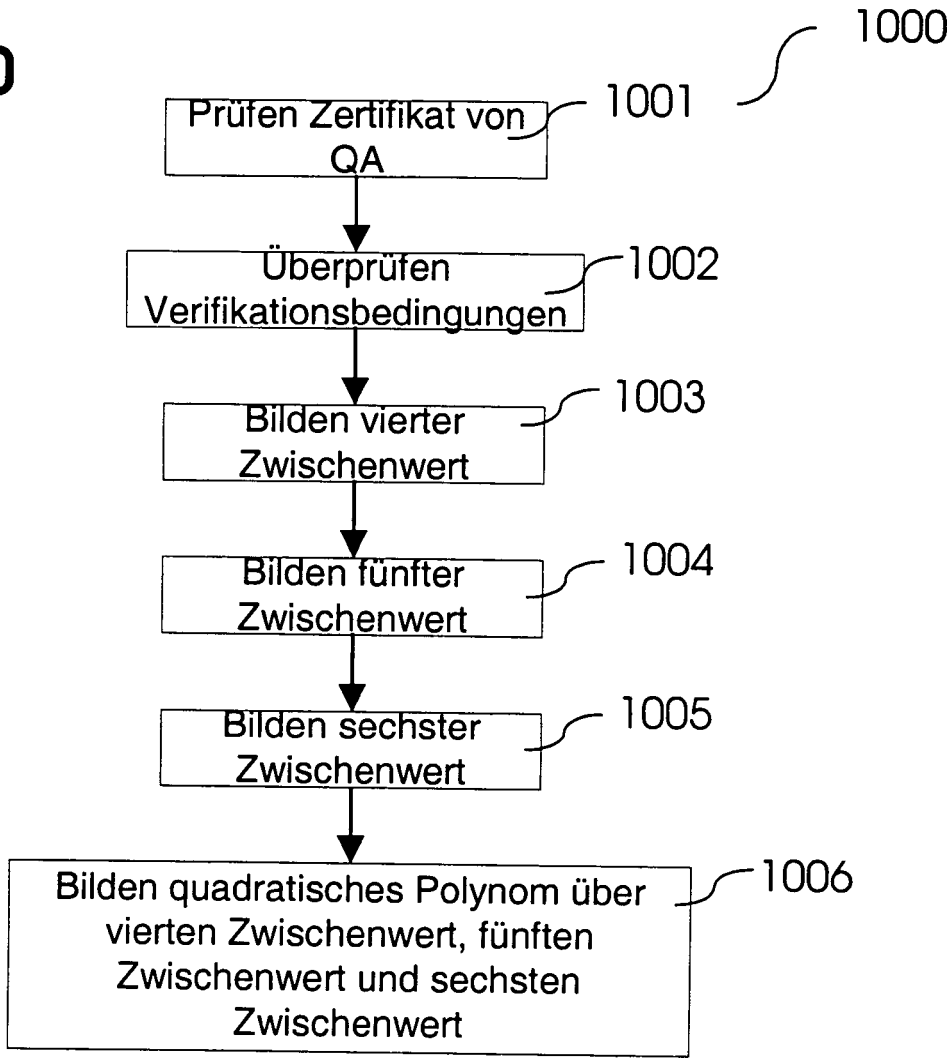

1000
Prüfen Zertifikat von QA
1001
Überprüfen Verifikationsbedingungen
1002
Bilden vierter Zwischenwert
1003
Bilden fünfter Zwischenwert
1004
Bilden sechster Zwischenwert
1005
Bilden quadratisches Polynom über vierten Zwischenwert, fünften Zwischenwert und sechsten Zwischenwert
1006

FIG 11

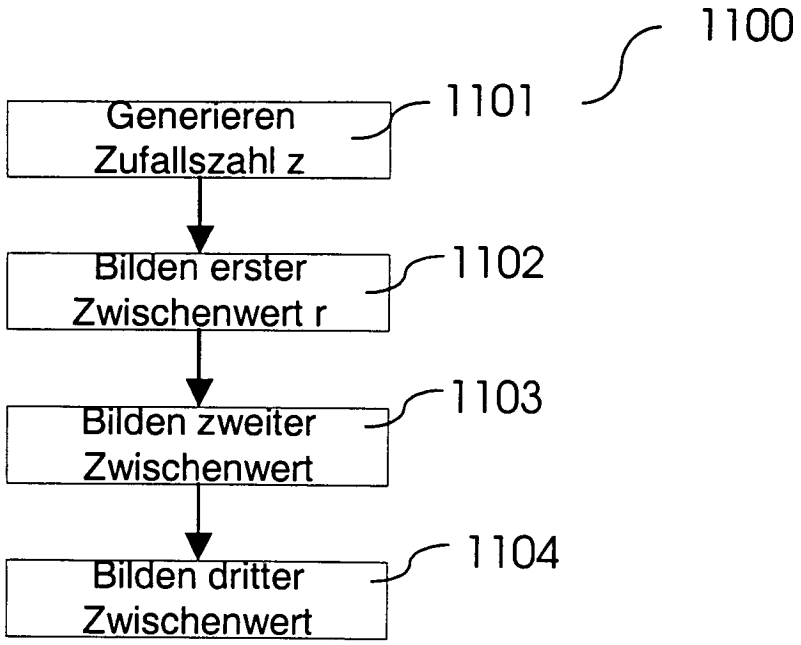

1100
Generieren Zufallszahl z
1101
Bilden erster Zwischenwert r
1102
Bilden zweiter Zwischenwert
1103
Bilden dritter Zwischenwert
1104

FIG 12

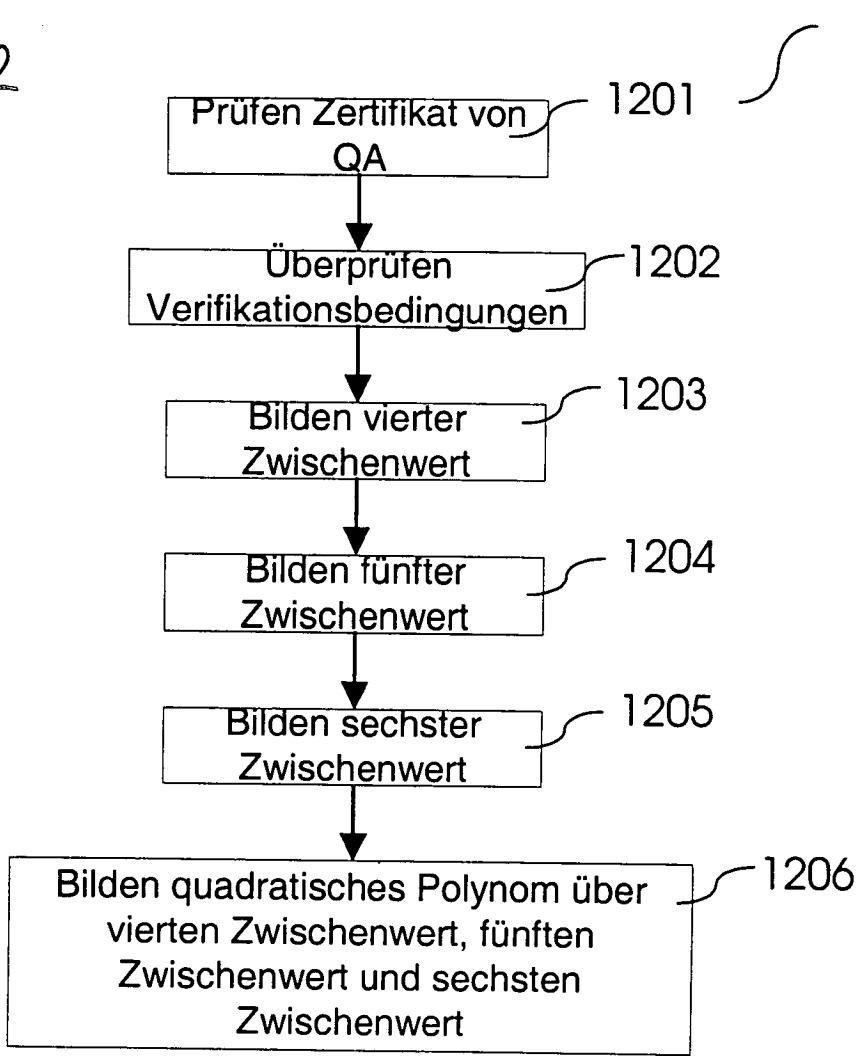

1200
Prüfen Zertifikat von QA
1201
Überprüfen Verifikationsbedingungen
1202
Bilden vierter Zwischenwert
1203
Bilden fünfter Zwischenwert
1204
Bilden sechster Zwischenwert
1205
Bilden quadratisches Polynom über vierten Zwischenwert, fünften Zwischenwert und sechsten Zwischenwert
1206

FIG 13

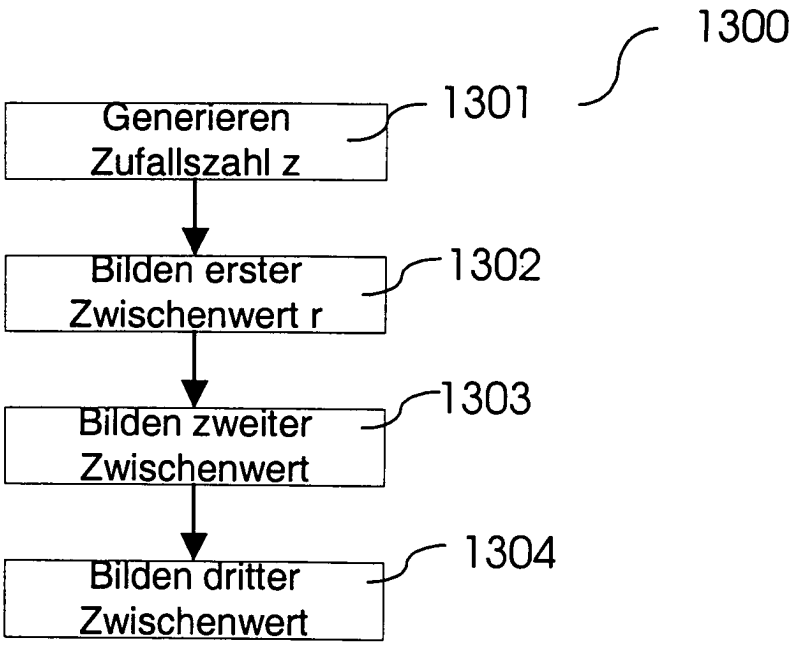
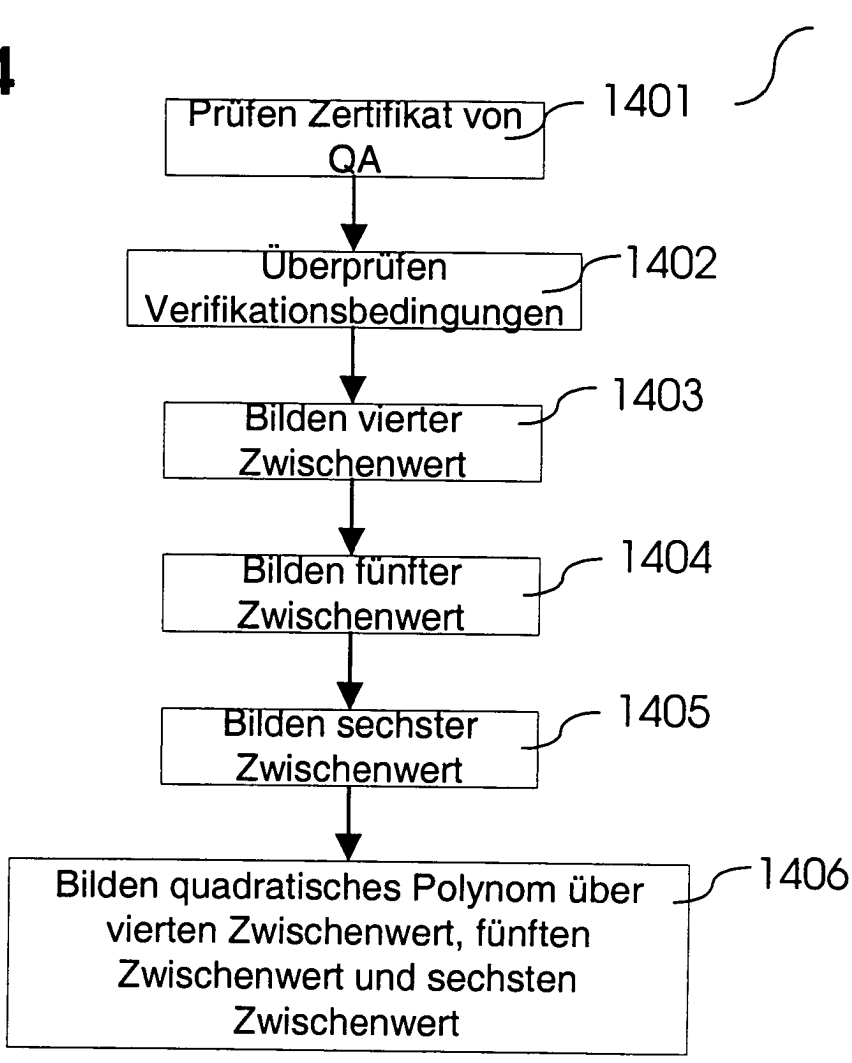

1300
Generieren Zufallszahl z
1301
Bilden erster Zwischenwert r
1302
Bilden zweiter Zwischenwert
1303
Bilden dritter Zwischenwert
1304

FIG 14


1400
Prüfen Zertifikat von QA
1401
Überprüfen Verifikationsbedingungen
1402
Bilden vierter Zwischenwert
1403
Bilden fünfter Zwischenwert
1404
Bilden sechster Zwischenwert
1405
Bilden quadratisches Polynom über vierten Zwischenwert, fünften Zwischenwert und sechsten Zwischenwert
1406