(19) Deutsches Patent- und Markenamt

(10) **DE 10 2018 128 561 A1** 2019.05.23

(12) # Offenlegungsschrift

(21) Aktenzeichen: **10 2018 128 561.2**
(22) Anmeldetag: **14.11.2018**
(43) Offenlegungstag: **23.05.2019**

(51) Int Cl.: ***G06F 21/60*** (2013.01)
G06F 21/14 (2013.01)
H04L 9/00 (2006.01)
G09C 1/00 (2006.01)

(30) Unionspriorität:
15/818,530 **20.11.2017** **US**

(71) Anmelder:
ANALOG DEVICES, INC., Norwood, Mass., US

(74) Vertreter:
WITTE, WELLER & PARTNER Patentanwälte mbB, 70173 Stuttgart, DE

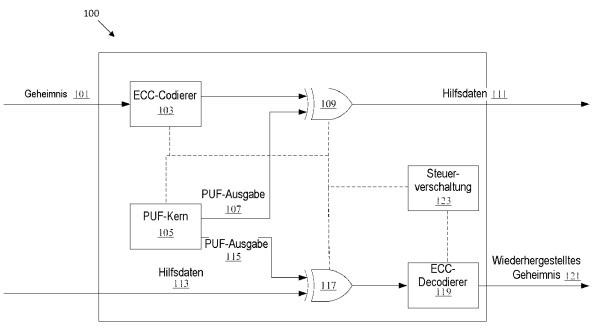
(72) Erfinder:
Poo, Tze Lei, Norwood, MA, US; Ahmad, Sadaf, Norwood, MA, US

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **EFFIZIENTE VERZÖGERUNGSBASIERTE PUF-IMPLEMENTIERUNG UNTER VERWENDUNG EINER OPTIMALEN WETTLAUFSTRATEGIE**

(57) Zusammenfassung: Nach verschiedenen Gesichtspunkten ist eine Vorrichtung mit einer verzögerungsbasierten physikalisch unklonbaren Funktion (PUF) vorgesehen. Nach einer Ausführungsform weist die PUF-Vorrichtung Verschaltung zum Generieren von Entropie-Ausgabebits durch Vergleichen oder „in einem Wettlauf Einsetzen“ einer Vielzahl von PUF-Zellen auf. Eine PUF-Zelle ist ein Baustein der PUF-Vorrichtung. Die PUF-Vorrichtung kann zum Beispiel zwei identisch konstruierte Schaltkreise mit nur prozessbedingten Schwankungen aufweisen und jeder Schaltkreis kann eine PUF-Zelle sein. Nach einem anderen Gesichtspunkt, falls PUF-Zellen mit einem gleichen Sieg- oder Niederlagen-Verlauf in einem Wettlauf verglichen werden, können Angreifer das Ergebnis des aktuellen Wettlaufs nicht aufgrund von vorangehenden Wettlaufergebnissen vorhersagen. Dementsprechend werden hierin Systeme und Verfahren zum Generieren von mehreren Wettlaufrunden auf Grundlage der vorangehenden Wettlaufrunden beschrieben. Deshalb kann eine PUF-Zelle in mehreren paarweisen Vergleichen verwendet werden, während eine maximale Entropie extrahiert wird.



Beschreibung

GEBIET DER OFFENBARUNG

[0001] Die vorliegende Offenbarung betrifft Codiersysteme und -verfahren mit einer physikalisch unklonbaren Funktion („PUF“).

STAND DER TECHNIK

[0002] Eine PUF kann eine Vorrichtung oder Verschaltung aufweisen, die eine Ausgabe generiert, die von eindeutigen physikalischen Eigenschaften der Vorrichtung abhängt. Schwankungen im Fertigungsprozess und in Teilen erzeugen zum Beispiel einen Chip, der elektrische Schaltkreise mit eindeutigen Hardwaremerkmalen aufweist, da auch die kleinsten Schwankungen (z. B. prozessabhängige Schwankungen bei Verzögerungen) für eine Eindeutigkeit sorgt.

ZUSAMMENFASSUNG DER OFFENBARUNG

[0003] Nach verschiedenen Gesichtspunkten ist eine Vorrichtung mit einer verzögerungsbasierten physikalisch unklonbaren Funktion (PUF) vorgesehen. Nach einer Ausführungsform weist die PUF-Vorrichtung Verschaltung zum Generieren von Entropie-Ausgabebits durch Vergleichen oder „in einem Wettlauf Einsetzen“ einer Vielzahl von PUF-Zellen auf. Die Wettläufe können zum Beispiel ausgeführt werden, indem die Frequenzen von zwei identisch konstruierten Ringoszillatoren (RO) oder zwei Pfade eines Arbiters verglichen werden. Eine PUF-Zelle ist ein Baustein der PUF-Vorrichtung. Die PUF-Vorrichtung kann zum Beispiel zwei identisch konstruierte Schaltkreise mit nur prozessbedingten Schwankungen aufweisen und jeder Schaltkreis kann eine PUF-Zelle sein. In einem Beispiel kann eine PUF-Zelle ein RO in einer RO-PUF, ein Verzögerungspfad in einer Arbiter-PUF, andere Komponenten von anderen verzögerungsbasierten PUFs oder eine beliebige Verschaltung zum Generieren der Ausgabe einer geeigneten PUF sein. Nach einem anderen Gesichtspunkt, falls PUF-Zellen mit einem gleichen Sieg- oder Niederlagen-Verlauf in einem Wettlauf verglichen werden, können Angreifer das Ergebnis des aktuellen Wettlaufs nicht aufgrund von vorangehenden Wettlaufergebnissen vorhersagen. Dementsprechend werden hierin Systeme und Verfahren zum Generieren von mehreren Wettlaufrunden auf Grundlage der vorangehenden Wettlaufrunden beschrieben. In einigen Ausführungsformen werden in jeder Runde die Sieger der vorangehenden Runde gegeneinander im Wettlauf eingesetzt, während die Verlierer im Wettlauf gegen Verlierer eingesetzt werden. Deshalb kann eine PUF-Zelle in mehreren paarweisen Vergleichen verwendet werden, während eine maximale Entropie extrahiert wird. Verschiedene Ausführungsformen bewahren die vollständige Entropie für Codierwerte, während die erforderliche Anzahl von PUF-Zellen im Vergleich zu herkömmlichen Ansätzen reduziert ist.

[0004] Nach einem Gesichtspunkt der vorliegenden Anmeldung ist ein Verfahren zum Generieren von Entropie in einer physikalisch unklonbaren Funktion (PUF) vorgesehen. Das Verfahren weist ein Zuordnen von ersten Paarungen von jeweiligen aus einer Vielzahl von PUF-Zellen in einer ersten Runde durch mindestens einen Prozessor; Generieren von jeweiligen ersten Ausgaben von jeder der Vielzahl von PUF-Zellen in der ersten Runde und Ermitteln von Ergebnissen für die erste Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den ersten Paarungen; Zuordnen von zweiten Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer zweiten, auf die erste Runde folgenden Runde; Generieren von jeweiligen zweiten Ausgaben von jeder der Vielzahl von PUF-Zellen in der zweiten Runde und Ermitteln von Ergebnissen für die zweite Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den zweiten Paarungen; und Generieren einer PUF-Ausgabe auf Grundlage der Ergebnisse der ersten Runde und der Ergebnisse der zweiten Runde auf.

[0005] Nach einer Ausführungsform weist das Verfahren ferner ein Verschleiern eines Geheimnisses unter Verwendung der PUF-Ausgabe auf. Nach einer Ausführungsform weist jede PUF-Zelle der Vielzahl der PUF-Zellen einen von einer Vielzahl von identisch konstruierten Schaltkreisen mit Unterschieden auf, die von Schwankungen im Fertigungsprozess herrühren. Nach einer Ausführungsform weist das Verfahren ferner ein Generieren eines Bits auf, das den Sieger jedes Paars von PUF-Zellen in jeder Runde repräsentiert, wobei ein Sieger eines Paars von PUF-Zellen mit einer Ausgabe eines binären Vergleichs der jeweiligen Ausgaben jeder PUF-Zelle im Paar assoziiert ist und wobei ferner die Paarungen der ersten Runde und der zweiten Runde eine Anzahl von Entropiebits generieren, die gleich der Anzahl der PUF-Zellen ist. Nach einer Ausführungsform weist das Verfahren ferner ein Anwenden eines Fehlerkorrekturcodes auf die jeweiligen ersten Ausgaben von jeder der Vielzahl der PUF-Zellen vor der zweiten Runde durch den mindestens einen Prozessor auf.

[0006] Nach einer Ausführungsform weist das Zuordnen von zweiten Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer zweiten, auf die erste Runde folgenden Runde auf: Zuordnen von Paarungen der jeweiligen Sieger aus jedem Paar von PUF-Zellen in der ersten Paarung; und Zuordnen von Paarungen der restlichen PUF-Zellen, die in der ersten Paarung keine Sieger waren. Nach einer Ausführungsform weist das Verfahren ferner ein Zuordnen, in einer auf die zweite Runde folgenden dritten Run-

de, von dritten Paarungen von jeweiligen der Vielzahl der PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde und der zweiten Runde auf, wobei jede PUF-Zelle der Vielzahl der PUF-Zellen mit einer PUF-Zelle der Vielzahl der PUF-Zellen gepaart wird, die der Sieger einer gleichen Anzahl von Runden (oder Paarungen) war; und Generieren von jeweiligen dritten Ausgaben von jeder der Vielzahl von PUF-Zellen in der dritten Runde und Ermitteln eines Siegers für jedes Paar von PUF-Zellen in den dritten Paarungen. Nach einer Ausführungsform wird ein erster Sieger eines ersten Paars in einer vorangehenden Runde mit einem zweiten Sieger eines zweiten Paars in der vorangehenden Runde gepaart. Nach einer Ausführungsform weist das Verfahren ferner ein Zuordnen zusätzlicher Runden von Paarungen auf Grundlage der Ergebnisse von vorangehenden Runden auf, wobei die Gesamtanzahl der Runden durch den mindestens einen Prozessor so eingeschränkt ist, dass sie den binären Logarithmus der Anzahl der PUF-Zellen in der Vielzahl der PUF-Zellen nicht überschreitet.

[0007] Nach einer Ausführungsform weist das Verfahren ferner ein Zuordnen zusätzlicher Runden von PUF-Zellenpaarungen auf Grundlage der Ergebnisse von vorangehenden Runden auf durch: Gruppieren von PUF-Zellen, die in einer vorangehenden Runde gepaart wurden, in eine Vielzahl von Gruppen; und Paaren jeder PUF-Zelle in einer ersten Gruppe mit einer jeweiligen PUF-Zelle in einer zweiten Gruppe.

[0008] Nach einem Gesichtspunkt der vorliegenden Anmeldung ist ein System zum Generieren von Entropie in einer physikalisch unklonbaren Funktion (PUF) vorgesehen. Das System weist eine Vielzahl von PUF-Zellen; und mindestens einen Prozessor auf, der ausgebildet ist: erste Paarungen von jeweiligen aus einer Vielzahl von PUF-Zellen in einer ersten Runde zuzuordnen; jeweilige erste Ausgaben von jeder der Vielzahl von PUF-Zellen in der ersten Runde zu generieren und Ergebnisse für die erste Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den ersten Paarungen zu ermitteln; zweite Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer zweiten, auf die erste Runde folgenden Runde zuzuordnen; jeweilige zweite Ausgaben von jeder der Vielzahl von PUF-Zellen in der zweiten Runde zu generieren und Ergebnisse für die zweite Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den zweiten Paarungen zu ermitteln; und eine PUF-Ausgabe auf Grundlage der Ergebnisse der ersten Runde und der Ergebnisse der zweiten Runde zu generieren.

[0009] Nach einer Ausführungsform ist der mindestens eine Prozessor ferner ausgebildet, ein Geheimnis unter Verwendung der PUF-Ausgabe zu verschleiern. Nach einer Ausführungsform weist jede PUF-Zelle der Vielzahl der PUF-Zellen einen von einer Vielzahl von identisch konstruierten Schaltkreisen mit Unterschieden auf, die von Schwankungen im Fertigungsprozess herrühren. Nach einer Ausführungsform weist das System ferner einen binären Komparator auf, wobei der mindestens eine Prozessor ferner ausgebildet ist, ein Bit zu generieren, das den Sieger jedes Paars von PUF-Zellen in jeder Runde repräsentiert, wobei ein Sieger eines Paars von PUF-Zellen mit einer Ausgabe des binären Komparators assoziiert ist, der jeweilige Ausgaben jeder PUF-Zelle im Paar verglichen hat, und wobei ferner die Paarungen der ersten Runde und der zweiten Runde eine Anzahl von Entropiebits generieren, die gleich der Anzahl der PUF-Zellen ist. Nach einer Ausführungsform wird ein erster Sieger eines ersten Paars in einer vorangehenden Runde mit einem zweiten Sieger eines zweiten Paars in der vorangehenden Runde gepaart, und wobei ferner der mindestens eine Prozessor ferner ausgebildet ist, vor einer aktuellen Runde einen Fehlerkorrekturcode auf jeweilige Ausgaben von jeder der Vielzahl der PUF-Zellen anzuwenden.

[0010] Nach einer Ausführungsform weist das Zuordnen von zweiten Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer zweiten, auf die erste Runde folgenden Runde ein Zuordnen von Paarungen der jeweiligen Sieger aus jedem Paar von PUF-Zellen in der ersten Paarung; und Zuordnen von Paarungen der restlichen PUF-Zellen auf, die in der ersten Paarung keine Sieger waren. Nach einer Ausführungsform ist der mindestens eine Prozessor ferner ausgebildet: in einer auf die zweite Runde folgenden dritten Runde dritte Paarungen von jeweiligen der Vielzahl der PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde und der zweiten Runde zuzuordnen, wobei jede PUF-Zelle der Vielzahl der PUF-Zellen mit einer PUF-Zelle der Vielzahl der PUF-Zellen gepaart wird, die der Sieger einer gleichen Anzahl von Runden (oder Paarungen) war; und jeweilige dritte Ausgaben von jeder der Vielzahl von PUF-Zellen in der dritten Runde zu generieren und einen Sieger für jedes Paar von PUF-Zellen in den dritten Paarungen zu ermitteln. Nach einer Ausführungsform ist der mindestens eine Prozessor ferner ausgebildet, zusätzliche Runden von Paarungen auf Grundlage der Ergebnisse von vorangehenden Runden zuzuordnen, wobei die Gesamtanzahl der Runden durch den mindestens einen Prozessor so eingeschränkt ist, dass sie den binären Logarithmus der Anzahl der PUF-Zellen in der Vielzahl der PUF-Zellen nicht überschreitet.

[0011] Nach einer Ausführungsform ist der mindestens eine Prozessor ferner ausgebildet, zusätzliche Runden von PUF-Zellenpaarungen auf Grundlage der Ergebnisse von vorangehenden Runden zuzuordnen durch: Gruppieren von PUF-Zellen, die in einer vorangehenden Runde gepaart wurden, in eine

Vielzahl von Gruppen; und Paaren jeder PUF-Zelle in einer ersten Gruppe mit einer jeweiligen PUF-Zelle in einer zweiten Gruppe.

[0012] Nach einem Gesichtspunkt der vorliegenden Anmeldung ist mindestens ein nichtflüchtiges computerlesbares Medium vorgesehen, das prozessorausführbare Anweisungen speichert. Die prozessorausführbaren Anweisungen bewirken, wenn sie ausgeführt werden, dass mindestens ein Prozessor ein Verfahren durchführt, aufweisend: Zuordnen von ersten Paarungen von jeweiligen aus einer Vielzahl von PUF-Zellen; Generieren von jeweiligen ersten Ausgaben von jeder der Vielzahl von PUF-Zellen in der ersten Runde und Ermitteln von Ergebnissen für die erste Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den ersten Paarungen; Zuordnen von zweiten Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer zweiten, auf die erste Runde folgenden Runde; Generieren von jeweiligen zweiten Ausgaben von jeder der Vielzahl von PUF-Zellen in der zweiten Runde und Ermitteln von Ergebnissen für die zweite Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den zweiten Paarungen; und Generieren einer PUF-Ausgabe auf Grundlage der Ergebnisse der ersten Runde und der Ergebnisse der zweiten Runde.

Figurenliste

[0013] Verschiedene Aspekte und Ausführungsformen der Offenbarung werden unter Bezugnahme auf die folgenden Figuren beschrieben. Es versteht sich, dass die Figuren nicht notwendigerweise maßstabgerecht gezeichnet sind. Elemente, die in mehreren Figuren erscheinen, sind in allen Figuren, in denen sie erscheinen, mit den gleichen Bezugszeichen versehen.

Fig. 1 zeigt ein veranschaulichendes Blockdiagramm einer PUF-basierten Kryptografievorrichtung, in der einige der hierin beschriebenen Technologien arbeiten können, nach einigen Ausführungsformen.

Fig. 2A zeigt ein veranschaulichendes Blockdiagramm einer PUF-Zelle, die verwendet wird, um eine PUF-Ausgabe zu generieren, nach einigen Ausführungsformen.

Fig. 2B zeigt ein veranschaulichendes Blockdiagramm einer PUF-Zelle, die verwendet wird, um eine PUF-Ausgabe zu generieren, nach einigen Ausführungsformen.

Fig. 3 zeigt ein veranschaulichendes Blockdiagramm eines PUF-Kerns zum Generieren einer PUF-Ausgabe nach einigen Ausführungsformen.

Fig. 4A-C zeigen veranschaulichende Runden einer Wettlaufreihenfolge für PUF-Zellen nach einigen Ausführungsformen.

Fig. 5A-B zeigen veranschaulichende Zeitgebungsdiagramme für Wettläufe von PUF-Zellen und zum Generieren von fehlerkorrigierten Ergebnissen, nach einigen Ausführungsformen.

Fig. 6 zeigt einen veranschaulichenden Prozessablauf für eine Festschreibungsphase zum Codieren eines Geheimnisses unter Verwendung einer PUF nach einigen Ausführungsformen.

Fig. 7 zeigt einen veranschaulichenden Prozessablauf für eine Wiederherstellungsphase zum Decodieren eines Geheimnisses unter Verwendung einer PUF nach einigen Ausführungsformen.

Fig. 8 zeigt einen veranschaulichenden Prozessablauf für Wettlaufrunden von PUF-Zellen, um eine PUF-Ausgabe zu generieren, nach einigen Ausführungsformen.

Fig. 9 zeigt einen veranschaulichenden Prozessablauf für ein Verfahren zum Generieren von Entropie in einer PUF-Codierung nach einigen Ausführungsformen.

AUSFÜHRLICHE BESCHREIBUNG

[0014] Nach verschiedenen Gesichtspunkten können herkömmliche Techniken zum Einsetzen von PUF-Zellen in Wettläufen, um Ausgabebits zu generieren, verbessert werden. Verschiedene Ausführungsformen ermöglichen zum Beispiel eine Reduktion in der Anzahl von erforderlichen PUF-Zellen, ohne eine Entropie zu reduzieren. Eine herkömmliche Technik unter Verwendung von verzögerungsbasierten PUFs sieht ein Vergleichen der Frequenz von benachbarten Paaren von Ringoszillatoren (RO) vor (auch als „im Wettlauf einsetzen“ bekannt), sodass jeder Ringoszillator nur einmal verglichen wird oder „an einem Wettlauf teilnimmt“. Derartige herkömmliche Ansätze erfordern eine Verwendung von zwei Ringoszillatoren, um ein Ausgabebit zu generieren. Bei einer zweiten Technik werden benachbarte Ringoszillatoren im Wettlauf eingesetzt, sodass jeder Ringoszillator zweimal an einem Wettlauf teilnimmt. Diese zweite Technik erfordert nur einen Ringoszillator für jedes Ausgabebit. Die von der zweiten Technik generierte Ausgabe generiert jedoch nicht die gesamte Entropie, da die Wahrscheinlichkeiten der Wettlaufergebnisse nicht unabhängig sind. Eine dritte Technik ermittelt eine Wettlaufreihenfolge auf Grundlage einer Eingabeherausforderung. Diese dritte Technik stellt nicht sicher, dass jeder Wettlauf die gesamte Entropie erzeugt, da die Ergebnisse jedes Wettlaufs nicht notwendigerweise unabhängig sind. Ferner kann diese dritte Technik gegenüber Angriffen mittels maschinellem Lernen oder anderen Fol-

gerungen ungeschützt sein, die die wahre Reihenfolge der Ringoszillatoren ermitteln. Unter derartigen Bedingungen können Angreifer das Ergebnis für eine beliebige Eingabeherausforderung vorhersagen.

[0015] Allgemein ausgedrückt kann eine effiziente Implementierung von einer physikalisch unklonbaren Funktion eine optimale Wettlaufstrategie (racing strategy) einsetzen, um eine bestimmte Anzahl von PUF-Zellen, wie Ringoszillatoren, für mehrere Runden auf sichere Weise gegeneinander wettlaufen zu lassen, wobei die maximale Entropie extrahiert wird. Nach verschiedenen Gesichtspunkten reduziert eine optimale Wettlaufstrategie die Anzahl der erforderlichen PUF-Zellen gegenüber herkömmlichen Ansätzen und bewahrt maximale Entropie. Nach einem Gesichtspunkt können die hierin beschriebenen Systeme und Verfahren einen minimalen Mehraufwand im Vergleich zu fixierten Wettlaufstrategien einsetzen, während die Anzahl der erforderlichen PUF-Zellen im Vergleich zu herkömmlichen Wettlaufstrategien reduziert ist. Obwohl einige Ausführungsformen in Bezug auf Ringoszillator-PUFs beschrieben sind, können andere Implementierungen unter Verwendung einer beliebigen geeigneten verzögerungsbasierten PUF-Implementierung, einschließlich beispielsweise einer Arbiter-PUF praktiziert werden.

[0016] Systeme und Verfahren zum Generieren von mehreren Wettlaufrunden auf Grundlage der vorangehenden Wettlaufrunden werden hierin beschrieben, um eine maximale Entropie in der Ausgabe jedes Wettlaufs zu bewahren. In einigen Ausführungsformen werden in jeder Runde die Sieger der vorangehenden Runde gegeneinander im Wettlauf eingesetzt, während die Verlierer im Wettlauf gegen Verlierer eingesetzt werden. Wie besprochen, falls die PUF-Zellen mit dem gleichen Sieg- oder Niederlagen-Verlauf in einem Wettlauf verglichen werden, kann man das Ergebnis des aktuellen Wettlaufs nicht aufgrund von vorangehenden Wettlaufergebnissen vorhersagen. Deshalb kann ein Ringoszillator in mehrfachen paarweisen Vergleichen verwendet werden. Nach verschiedenen Ausführungsformen schränkt eine optimale Wettlaufstrategie die Anzahl der Wettlaufrunden ein, die Ergebnisse mit der gesamten Entropie ergeben. Wettläufe, die über derartige Schwellenwerte hinaus ausgeführt werden, riskieren eine Vorhersagbarkeit auf Grundlage von vorangehenden Ergebnissen. Nach einer Ausführungsform führt das System eine Fehlerkorrektur zusammen mit mehreren Wettlaufrunden durch, um Probleme in Bezug auf Fehlerfortpflanzung zu vermeiden. PUF-Ausgaben sind zum Beispiel als verrauscht bekannt - deshalb kann eine Fehlerkorrektur eingesetzt werden, um konsistente Ausgaben zu erzielen. In einer Implementierung ist das System ausgebildet, ein Ergebnis von jeder Wettlaufrunde zu generieren und einer Fehlerkorrektur daran auszuführen. Auf Grundlage der Ausführung der Fehlerkorrektur während des Abschlusses mindestens einer Wettlaufrunde propagieren die korrigierten Ergebnisse keine Fehler in nachfolgende Runden (z. B. durch falsches Identifizieren eines Siegers oder Verlierers). In anderen Beispielen kann jede Wettlaufrunde vor dem Zuordnen von Paarungen für die nachfolgende Runde fehlerkorrigiert werden und die Fortpflanzung von Fehlern durch nachfolgende Runden verhindern.

[0017] Einige Ausführungsformen der hierin beschriebenen Technologie beheben einige der oben besprochenen Nachteile herkömmlicher Technologie zum Generieren von PUFs. Nicht jede Ausführungsform muss jeden dieser Nachteile oder die oben besprochenen Verbesserungen beheben, und einige Ausführungsformen beheben möglicherweise keine der Nachteile. Als solches sollte klar sein, dass Gesichtspunkte der hierin beschriebenen Technologie nicht darauf beschränkt sind, alle oder beliebige der oben besprochenen Nachteile von herkömmlichen PUF-Systemen zu beheben.

[0018] **Fig. 1** zeigt ein veranschaulichendes Blockdiagramm einer PUF-basierten Kryptografievorrichtung **100**, in der einige der hierin beschriebenen Technologien arbeiten können, nach einigen Ausführungsformen. **Fig. 1** veranschaulicht ein Geheimnis **101** (z. B. einen privaten Verschlüsselungsschlüssel, ein Geheimnis, eine Zufallszahl, unter anderen Datenwerten), eine PUF-Ausgabe (z. B. eine Anzahl von Bits, die sich aus einem Ermitteln einer PUF-Ausgabe ergeben) **107**, Hilfsdaten **111**, Hilfsdaten **113**, eine PUF-Ausgabe **115** und ein wiederhergestelltes Geheimnis **121** (z. B. eine Wiederherstellung des Geheimnisses auf Grundlage der Hilfsdaten), die von einem Fehlerkorrekturcode(„ECC“)-Codierer **103**, einem PUF-Kern (z. B. Verschaltung, die mehrere PUF-Zellen aufweist) **105**, einem Operator **109**, einen Operator **117**, einem ECC-Decodierer **119** und Steuerverschaltung **123** generiert, verarbeitet und/oder gesteuert werden. Die Kryptografievorrichtung **100** kann betriebsfähig und/oder ausgebildet sein, beliebige geeignete Eingabewerte zu verschlüsseln, zu codieren, zu verschleiern, erneut zu generieren und/oder wiederherzustellen oder derartige kryptografische Operationen an beliebigen Daten in Übereinstimmung mit den hierin beschriebenen Systemen und Verfahren durchzuführen.

[0019] In der veranschaulichten Ausführungsform kann die Vorrichtung **100** verwendet werden, um ein Geheimnis **101** zu verschlüsseln und zu entschlüsseln. Das Geheimnis kann eine beliebige Folge von Bits sein, die für die Eingabe in die Vorrichtung **100** geeignet ist. Das Geheimnis **101** wird der Vorrichtung **100** während einer Festschreibphase bereitgestellt, die unten beschrieben wird, und in einigen Ausführungsformen können mehrere Geheimnisse unter Verwendung der Vorrichtung **100** eindeutig codiert und/oder decodiert werden. Das Geheimnis **101** kann

ein privater Verschlüsselungsschlüssel, eine Zufallszahl, eine Seriennummer, ein Geheimnis, sensible Informationen oder beliebige andere Daten sein, die privat gehalten werden sollen, einschließlich, zum Beispiel, Anteile eines Geheimnisses oder Schlüssels. In einigen Ausführungsformen kann das Geheimnis **101** die Ausgabe einer PUF sein. In weiteren Ausführungsformen kann die PUF-Ausgabe **107** anstelle des Geheimnisses **101** verwendet werden.

[0020] Der ECC-Codierer **103** empfängt das Geheimnis **101** und wendet einen Fehlerkorrekturcode an, um ECC-Daten zum Korrigieren von Fehlern im Geheimnis **101** und dem wiederhergestellten Geheimnis **121** zu berechnen. In einem Beispiel sorgt das wiederhergestellte Geheimnis für eine erneute Generierung des codierten Geheimnisses, insbesondere ohne dass das Geheimnis auf der Vorrichtung gespeichert werden muss. Stattdessen sorgt die Abbildung einer PUF-Ausgabe mit einem Hilfswert auf das Geheimnis für eine sichere erneute Generierung zu einem späteren Zeitpunkt. Der ECC-Codierer kann ausgebildet sein, unter Verwendung eines beliebigen geeigneten ECC-Codes ECC-Daten zu berechnen und eine beliebige Anzahl von Bitfehlern zum erfolgreichen Wiederherstellen des Geheimnisses korrigieren. In einigen Ausführungsformen führt der ECC-Codierer eine Bose-Chaudhuri-Hocquenghem(BCH)-Codierung durch. Der ECC-Decodierer **119** kann ein beliebiger ECC-Decodierer oder eine beliebige Verarbeitungsverschaltung sein, die ausgebildet ist, Fehler unter Verwendung der vom ECC-Codierer **103** implementierten ECC zu decodieren und zu erkennen/korrigieren. Der ECC-Codierer **103** und der ECC-Decodierer **119** können ausgebildet sein, Fehler im PUF-Kern **105** zu korrigieren, wenn die PUF-Ausgabe ermittelt wird. In einem Beispiel wird der ECC-Codierer **103** verwendet, um Hilfsdaten zum Abbilden einer PUF-Antwort auf ein Geheimnis zu generieren. In weiteren Ausführungsformen können der ECC-Codierer **103** und der ECC-Decodierer **119** ausgebildet sein, Bitfehler zu erkennen und zu korrigieren, die sich aus Unterschieden in den PUF-Ausgaben **107** und **115** ergeben, die verwendet werden, um das Geheimnis zu codieren und erneut zu generieren. Auch wenn zum Beispiel eine zum Decodieren verwendete PUF-Ausgabe verrauscht ist, kann der ECC-Decodierer **119** zuverlässig ein Geheimnis erneut generieren, solange die PUF-Ausgabe innerhalb einer korrigierbaren Anzahl von Bitfehlern (die z. B. von den ECC-Codierparametern bestimmt wird) vom codierten Wert liegt, wie er von den Hilfsdaten bereitgestellt wird.

[0021] Der PUF-Kern **105** kann verwendet werden, um PUF-Ausgaben zum Verschlüsseln und Entschlüsseln des Geheimnisses **101** zu generieren. Der PUF-Kern **105** kann Verschaltung aufweisen, die eine Ausgabe generiert, die von eindeutigen physikalischen Eigenschaften von einer oder mehreren PUF-Zellen im PUF-Kern **105** abhängt. Schwankungen in Fertigungsprozessen und in Teilen können zum Beispiel einen Chip erzeugen, der elektrische Schaltkreise mit eindeutigen Hardwaremerkmalen aufweist. Der PUF-Kern **105** kann einen oder mehrere elektrische Schaltkreise aufweisen, die Ausgaben generieren, die auf den eindeutigen Hardwaremerkmalen basieren, die für den einen oder die mehreren elektrischen Schaltkreise spezifisch sind. Beispiele von PUF-Kerntypen weisen Arbiter-PUFs, RO-PUFs, Butterfly-PUFs, andere verzögerungsbasierte PUFs oder eine beliebige PUF-Implementierung, die Werte von identisch konstruierten Schaltkreisen vergleicht, auf. In einigen Ausführungsformen wird der PUF-Kern **105** zum Beispiel von der Steuerverschaltung **123** angewiesen, eine PUF-Ausgabe (z. B. **107** oder **115**) zu generieren und erzeugt eine PUF-Ausgabe, die nicht auf irgendeinem Eingabewert in den PUF-Kern **105** basiert. Der PUF-Kern **105** kann zum Beispiel einen Antwortwert auf Grundlage von Wettlaufpaaren von PUF-Zellen ohne eine besondere Herausforderungsanforderung zurückgeben.

[0022] In einigen Ausführungsformen kann der PUF-Kern **105** ausgebildet sein, eine Herausforderungseingabe zu empfangen, die verwendet wird, um anfängliche PUF-Zellenpaarungen anzugeben, um die PUF-Ausgabe zu berechnen. In einigen Beispielen kann die Steuerverschaltung **123** die erste Runde von Paarungen angeben oder ein anfänglicher Paarungssatz kann vordefiniert sein. In weiteren Ausführungsformen gibt die Herausforderungseingabe die erste Paarungsrunde indirekt an, indem sie von einer begrenzten Anzahl von gültigen Eingaben auf Paarungen der ersten Runde abbildet, die für die Verschlüsselungsvorrichtung geheim bleiben.

[0023] PUF-Ausgaben **107** und **115** sind Antworten, die vom PUF-Kern und/oder den PUF-Zellen erhalten wurden (z. B. Zeichenfolgen von Bits, die für ein Verschleiern des Geheimnisses **101** geeignet sind). In einigen Ausführungsformen weist jede PUF-Ausgabe eine Anzahl von Bits (z. B. 32, 64, 128 oder 256 Bits) von Entropie auf, die durch ein Wettlaufen von Paaren von PUF-Zellen erhalten wurden. In einigen Ausführungsformen repräsentiert jedes Bit in der PUF-Ausgabe ein volles Entropiebit. Die PUF-Ausgaben **107** und **115** können bei Fehlen von Fehlern während der PUF-Ausgabengenerierung identisch sein. Die PUF-Ausgaben **107** und **115** können unterschiedlich sein, zum Beispiel, da PUFs verrauscht sein können, aber das Geheimnis **101** kann möglicherweise erfolgreich wiederhergestellt werden, falls sich die PUF-Ausgaben **107** und **115** nicht um mehr als eine Maximalanzahl von Bits unterscheiden, die von einem ECC korrigiert werden kann (z. B. einem ECC, der vom ECC-Codierer **103** und vom ECC-Codierer **119** implementiert werden kann). In einigen Ausführungsformen wird eine PUF-Ausgabe **107** während der Festschreibphase verwendet, wobei die Fehlerkor-

rekturcodedaten an das Geheimnis **101** angehängt werden, um Hilfsdaten **111** für die nachfolgende erneute Generierung des geheimen Werts während der Wiederherstellungsphase mit der PUF-Ausgabe **115** zu erzeugen. In einigen Ausführungsformen wird der ECC-Decodierer **119** verwendet, um einen Abschnitt des wiederhergestellten Geheimnisses **121** zu decodieren, das unter Verwendung eines Abschnitts der PUF-Ausgabe **115** wiederhergestellt wurde - wobei zum Beispiel eine aus einer Wettlaufrunde abgeleitete Ausgabe korrigiert wird. In weiteren Ausführungsformen werden erkannte und/oder korrigierte Fehler an den PUF-Kern **105** zum Generieren weiterer Abschnitte der PUF-Ausgabe **115** kommuniziert und mehrere korrigierte Abschnitte können zur erneuten Generierung des Geheimnisses kombiniert werden. Die Fehlerkorrektur der PUF-Ausgabe wird unter Bezugnahme auf **Fig. 5A-B** besprochen.

[0024] In einigen Ausführungsformen können die PUF-Ausgaben (z. B. **107** und **115**) als ein Geheimnis verwendet werden. Eine PUF-Ausgabe kann zum Beispiel als eine eindeutige Vorrichtungskennung, als ein Verschlüsselungsschlüssel, als ein unter Verwendung einer zweiten PUF-Ausgabe zu verschlüsselndes oder verschleierndes Geheimnis oder für einen beliebigen anderen geeigneten Zweck verwendet werden. In einigen Ausführungsformen kann die PUF-Ausgabe **107** anstelle des Geheimnisses **101** verwendet werden und in den ECC-Codierer **103** eingegeben werden. In weiteren Ausführungsformen können die auf Grundlage der PUF-Ausgabe (z. B. **107**) generierten ECC-Daten als die Hilfsdaten **111** zum Wiederherstellen der geheimen PUF-Ausgabe gespeichert werden.

[0025] In einigen Ausführungsformen wird jede Ausgabe des PUF-Kerns **105** in Runden generiert. In einigen Ausführungsformen wird die Generierung einer PUF-Ausgabe durch die Steuerverschaltung **123** oder einen Prozessor (z. B. eine CPU, ein FPGA usw.) gesteuert. In einigen Ausführungsformen kann die Steuerverschaltung (z. B. **123**) oder der Prozessor im PUF-Kern **105** enthalten sein.

[0026] In einigen Ausführungsformen werden in einer ersten Runde jeweilige Paarungen einer Vielzahl von PUF-Zellen im PUF-Kern **105** zugeordnet. In einigen Ausführungsformen kann die erste Paarung eine beliebige geeignete Abbildung von Paaren sein, die zum erneuten Generieren der PUF-Ausgabe reproduziert werden kann. Falls die Abbildung von Paaren, die verwendet wird, um die PUF-Ausgabe **107** zu generieren, für die erste Paarung zum Generieren der PUF-Ausgabe **115** verwendet wird, können die resultierenden PUF-Ausgaben **107** und **115** übereinstimmen (z. B. wie sie auf Fehler korrigiert wurden). Falls die Abbildung von Paaren der ersten Runde geändert wird, stimmen die Ausgaben wahrscheinlich nicht überein.

[0027] In einem Beispiel kann die erste Paarungsrunde benachbarte PUF-Zellen paaren. In einigen Ausführungsformen sind die PUF-Zellen geordnet und/oder von 1 bis N nummeriert. In der ersten Runde können die ungerade nummerierten PUF-Zellen mit einer benachbarten gerade nummerierten PUF-Zelle gepaart werden, wobei z. B. die PUF-Zellen **1** und **2**, die PUF-Zellen **3** und **4**, die PUF-Zellen **5** und **6**, ... und die PUF-Zellen N-1 und N gepaart werden. In weiteren Ausführungsformen werden die PUF-Zellen mit einer verfügbaren PUF-Zelle gepaart, die um N/2 Plätze oder eine andere geeignete Distanz getrennt ist. In anderen Ausführungsformen repräsentiert die erste Runde von PUF-Zellen-Paarungen eine beliebige paarweise Abbildung von PUF-Zellen.

[0028] Jedes Paar von PUF-Zellen in der ersten Paarung kann im Wettlauf gegeneinander eingesetzt werden, indem bewirkt und/oder ermöglicht wird, dass jede PUF-Zelle im Paar eine Ausgabe generiert. In einigen Ausführungsformen wird eine PUF-Zelle jedes Paars auf Grundlage eines binären Vergleichs der Ausgaben als der Sieger ermittelt und eine PUF-Zelle jedes Paars wird als der Verlierer ermittelt. Wenn die PUF-Zellen zum Beispiel Ringoszillatoren aufweisen, können Ausgabeübergänge jedes Oszillators für eine Zeitperiode gezählt und verglichen werden, wobei der Oszillator mit der höheren Frequenz einen größeren Zählerstand erzeugt und als der Sieger ermittelt wird. In einigen Ausführungsformen kann eine schiedsrichterbasierte PUF eine Schiedsrichterausgabe mit jedem von zwei Verzögerungspfaden assoziieren. Es sollte klar sein, dass Komponenten anderer verzögerungsbasierten PUFs als geeignete PUF-Zellen verwendet werden können.

[0029] Wenn PUF-Zellen im Wettlauf gegeneinander eingesetzt werden, die vorher noch nicht miteinander und/oder einem gemeinsamen Gegner gepaart worden sind, kann ein volles Entropiebit durch das Resultat des Wettlaufs generiert werden, da es keine Apriori-Informationen gibt, die verwendet werden können, um das Resultat vorherzusagen. In einigen Ausführungsformen werden N PUF-Zellen in N/2 Paaren pro Runde im Wettlauf eingesetzt. Deshalb können in jeder Runde, in der PUF-Zellen ohne gemeinsame Gegner gepaart werden, N/2 Entropiebits generiert werden. In zwei Runden können N PUF-Zellen N Entropiebits generieren. In vier Runden können N PUF-Zellen 2*N Entropiebits generieren (z. B. können 64 PUF-Zellen in vier Runden **128** Entropiebits generieren). Nach einigen Ausführungsformen, um zu garantieren, dass jeder Wettlauf ein volles Entropiebit erzeugt, kann die Anzahl der Paarungsrunden auf den binären Logarithmus der Anzahl der PUF-Zellen, N, eingeschränkt werden. In einem Beispiel ist nach diesem Punkt die Wettlaufreihenfolge der PUF-Zellen großteils ermittelt und zukünftige Wettläufe ergeben jeweils weniger als ein volles Entropiebit. Es sollte klar sein, dass das Ausmaß, in dem die Wettlaufrei-

henfolge nach einer bestimmten Anzahl von Runden ermittelt ist, auf Grundlage von Prozessunterschieden variieren kann, die die Unterschiede in PUF-Zellenausgaben bestimmen, und für jede Vorrichtung einzigartig sein kann. In einigen Ausführungsformen können Wettlaufrunden nach dem binären Logarithmus der Anzahl der PUF-Zellen fortfahren, mit einem möglichen Verlust von Entropie.

[0030] Die Paarungen der PUF-Zellen können durch den PUF-Kern **105** und/oder die Steuerverschaltung **123** ausgewählt werden, um sicherzustellen, dass gepaarte PUF-Zellen keinen gemeinsamen vergangenen Gegner aufweisen. In einigen Ausführungsformen wird die Auswahl aktiv durchgeführt, zum Beispiel durch Nachverfolgung des Satzes von Gegnern, mit dem jede PUF-Zelle konfrontiert wird, und Auswählen von Paaren, dessen Sätze von Gegnern einen Schnitt von null aufweisen. In weiteren Ausführungsformen gruppiert die Wettlaufreihenfolge die PUF-Zellen, um gemeinsame vergangene Gegner zu verhindern.

[0031] In einigen Ausführungsformen wird die Wettlaufreihenfolge verwendet, um Bedingungen dafür zu schaffen, dass durch jede Paarung und/oder jeden Wettlauf eine volle Entropie generiert wird. In der ersten Runde können PUF-Zellen keinesfalls gemeinsame Gegner aufweisen, da keine Wettläufe durchgeführt worden sind. Jeder Sieger in der ersten Runde kann mit einem anderen Sieger von der ersten Runde gepaart werden und jeder Verlierer wird mit einem anderen Verlierer gepaart. In einer zweiten Runde, die auf diese Weise zugeordnet wird, kann kein PUF-Zellenpaar einen überschneidenden Verlauf von Gegnern aufweisen, da jeder Sieger (Verlierer) vorher genau gegen einen jeweiligen Verlierer (Sieger) im Wettlauf angetreten ist. In einigen Ausführungsformen wird eine dritte Runde von Paarungen so zugeordnet, dass Sieger (Verlierer) der vorangehenden Runde mit Siegern (Verlierern) der vorangehenden Runde gepaart werden. In einigen Ausführungsformen werden PUF-Zellen gepaart, die den gleichen Verlauf aufweisen und die Wettläufe in der gleichen Reihenfolge gewonnen oder verloren haben. In einigen Ausführungsformen werden PUF-Zellen maximal oder genau einmal pro Runde im Wettlauf eingesetzt. Das maximal einmalige Einsetzen jeder PUF-Zelle im Wettlauf pro Runde kann sicherstellen, dass jeder Wettlauf ein volles Entropiebit erzeugt.

[0032] Verschiedene Ausführungsformen von Systemen und Verfahren für PUF-basierte Codierung werden weiter unter Bezugnahme auf **Fig. 3**, **Fig. 4A-C**, **Fig. 5A-B**, **Fig. 8** und **Fig. 9** beschrieben.

[0033] In einigen Ausführungsformen kann eine Ausgabe des PUF-Kerns **105** verrauscht sein. Die Unterschiede zwischen einigen der PUF-Zellen können zum Beispiel hinreichend gering sein, sodass die Zellen als schwach angesehen werden und während einer PUF-Ausgabengenerierung fehleranfällig sein. In einigen Ausführungsformen können die PUF-Zellen im PUF-Kern **105** temperaturempfindliche Ausgaben aufweisen. Die Frequenz von Ringoszillator-PUF-Zellen kann zum Beispiel mit unterschiedlichen Raten als eine Funktion der Temperatur variieren und es kann sein, dass schwache Zellen bei einer bestimmten Temperatur überkreuzte Frequenzen aufweisen. In weiteren Ausführungsformen wird dem PUF-Kern **105** ein Konfigurationssignal bereitgestellt, um zuverlässigere und robustere PUF-Zellen zu erzeugen. Das Konfigurationssignal kann zum Beispiel zum Zeitpunkt der Fertigung verwendet werden, um die Ausgaben der PUF-Zellen zu konfigurieren, sodass sie bei Betriebstemperaturen hinreichend getrennt bleiben. In einigen Ausführungsformen sind die PUF-Zellen auf Grundlage der Paarung in der ersten Runde konfiguriert, um ein zuverlässigeres Wettlaufergebnis bereitzustellen. Beispiele des Konfigurationssignals und der Konfiguration der PUF-Zellen werden unter Bezugnahme auf **Fig. 2A** besprochen.

[0034] Nach einer Ausführungsform wird die Ausgabe des PUF-Kerns **105** vom Operator **109** empfangen. Der Operator **109** kann ein beliebiger Operator (z. B. ein bitweises XOR-Gatter oder eine arithmetisch-logische Einheit, unter anderen Schaltkreisen) zum Verwenden der PUF **107** sein, um das Geheimnis **101** so zu verschleiern, dass die Hilfsdaten **111** nicht verwendet werden können, um das Geheimnis **101** ohne die korrekte PUF-Ausgabe zu ermitteln. Der Operator **109** kann eine beliebige umkehrbare Operation sein, um eine Wiederherstellung des Geheimnisses **101** aus den Hilfsdaten **111** und einer PUF-Ausgabe zu ermöglichen. In einigen Ausführungsformen ist der Operator **109** eine bitweise XOR-Operation, die ihr eigenes Inverses sein kann. In einigen Ausführungsformen können die Operatoren **109** und **117** beide bitweise XOR-Operationen sein. In einigen Ausführungsformen ist der Operator **109** eine Maskierungsoperation, die die Ausgabebits ändert, die Bits im Geheimnis **101** auf Grundlage der PUF-Ausgabe **107** entsprechen. In weiteren Ausführungsformen ist der Operator **109** ein kryptografischer Codierer und der Operator **117** ist ein entsprechender Decodierer. Der Operator **117** wird verwendet, um die vom Operator **109** durchgeführte Verschleierung des Geheimnisses **101** aufzuheben, um das Geheimnis **101** wiederherzustellen.

[0035] Das wiederhergestellte Geheimnis **121** kann aus den Hilfsdaten **113** wiederhergestellt werden. Das wiederhergestellte Geheimnis **121** stimmt mit dem Geheimnis **101** überein, falls die Hilfsdaten **113** mit den Hilfsdaten **111** übereinstimmen. In einigen Ausführungsformen wird vorweggenommen, dass die Hilfsdaten **113** mit den Hilfsdaten **111** übereinstimmen, aber es sollte klar sein, dass Vorrichtungsfehler oder ein Angreifer bewirken können, dass

sich die Hilfsdaten **113** von den Hilfsdaten **111** unterscheiden. In diesem Fall würde das Geheimnis nicht wiederhergestellt. In weiteren Ausführungsformen kann die Vorrichtung mehr als ein Geheimnis codieren, wobei jedes Geheimnis festgeschrieben und auf jeweilige Hilfsdaten abgebildet wird, und jedes Geheimnis kann nur unter Verwendung der jeweils korrekten Hilfsdaten erneut generiert werden. Um das wiederhergestellte Geheimnis **121** zu generieren (z. B., um das Geheimnis **101** zu entschlüsseln), generiert der PUF-Kern **105** eine zweite PUF-Ausgabe **115**. In einigen Ausführungsformen arbeitet der PUF-Kern **105** auf einer Vergleichsbasis zwischen PUF-Zellen (z. B. auf Grundlage von Paarungen, die von der Steuerverschaltung **123** zugeordnet wurden), und der Wert jeder Ausgabe des PUF-Kerns **105** kann von irgendeiner Eingabe in den PUF-Kern **105** unabhängig sein. Es kann erwartet werden, dass die PUF-Ausgaben **107** und **115** gleich sind, nachdem sie mit einem Fehlerkorrekturcode decodiert wurden. Das wiederhergestellte Geheimnis **121** kann ohne Enthüllen irgendeiner PUF-Ausgabe generiert werden.

[0036] Die Steuerverschaltung **123** kann eine beliebige geeignete Verarbeitungsverschaltung sein, wie ein Mikroprozessor, ein feldprogrammierbares Gatearray (FPGA), ein anwendungsspezifischer integrierter Schaltkreis (ASIC) oder eine beliebige andere geeignete Verschaltung zum Steuern der Generierung der PUF-Ausgabe, der Zeitgebung und der Paarungen. In einigen Ausführungsformen wird die Steuerverschaltung verwendet, um zu steuern, ob sich die Vorrichtung **100** in einem bestimmten Konfigurationsmodus befindet. In einem Beispiel weisen die Konfigurationsmodi eine Festschreibungsphase zum Codieren oder Binden eines Geheimnisses an jeweilige Hilfsdaten oder eine Wiederherstellungsphase zum erneuten Generieren eines gebundenen Geheimnisses auf.

[0037] In einigen Ausführungsformen generiert die Verschlüsselungsvorrichtung während einer Festschreibungsphase Hilfsdaten aus dem Geheimnis **101**. In einigen Beispielen können die codierten Hilfsdaten **111** sicher öffentlich gemacht werden. In anderen Beispielen können die Hilfsdaten **111** Operatoren oder Vorrichtungen bereitgestellt werden, die an PUF-basierten Codieroperationen teilnehmen. In einigen Ausführungsformen ist die Vorrichtung **100** durch die Steuerverschaltung **123** ausgebildet, sich in der Festschreibphase zu befinden. In einigen Ausführungsformen empfängt die Steuerverschaltung **123** ein externes Signal, um die Festschreibphase zu beginnen.

[0038] Während der Festschreibphase wird das Geheimnis **101** unter Verwendung einer PUF verschlüsselt, codiert und/oder verschleiert. Die Verschlüsselungsvorrichtung erhält das Geheimnis **101** unter Verwendung einer beliebigen geeigneten Eingabe. Der ECC-Codierer **103** wendet einen ECC auf das Geheimnis **101** an, um ECC-Daten zu erzeugen, die zur weiteren Verarbeitung an das Geheimnis **101** angehängt werden. Der PUF-Kern **105** generiert eine PUF-Ausgabe **107**. Die PUF-Ausgabe **107** kann in einen ECC-Codierer (z. B. **103**) eingegeben werden, um Fehlerkorrekturdaten für eine zukünftige PUF-Ausgabengenerierung bereitzustellen. Der Operator **109**, z. B. XOR, wird auf die PUF-Ausgabe und das Geheimnis **101** mit angehängten ECC-Daten angewandt, um die Hilfsdaten **111** zu generieren.

[0039] Während der Wiederherstellungsphase wird das wiederhergestellte Geheimnis **121** aus den Hilfsdaten generiert. Die Verschlüsselungsvorrichtung erhält die Hilfsdaten **113** unter Verwendung einer beliebigen geeigneten Eingabe. Der PUF-Kern **105** generiert die PUF-Ausgabe **115**. In einigen Ausführungsformen stellen Fehlerkorrekturcodes sicher, dass die PUF-Ausgabe **115** mit der PUF-Ausgabe **107** übereinstimmt, die in der Festschreibphase verwendet wird. Der Operator **117** führt das Inverse des Operators **109** durch, der verwendet wurde, um das Geheimnis **101** während der Festschreibphase zu verschleiern, um eine Schätzung des Geheimnisses **101** zu generieren. Der ECC-Decodierer **119** decodiert das geschätzte Geheimnis und die ECC-Daten, die während der Festschreibphase angehängt wurden, um das wiederhergestellte Geheimnis **121** zu erzeugen. Nach einem Beispiel, vorausgesetzt, die Hilfsdaten **113** stimmen mit den Hilfsdaten **111** überein, stimmt das wiederhergestellte Geheimnis **121** mit dem Geheimnis **101** überein.

[0040] **Fig. 2A** zeigt ein veranschaulichendes Blockdiagramm einer PUF-Zelle **230**, die verwendet wird, um eine PUF-Ausgabe zu generieren, nach einigen Ausführungsformen. **Fig. 2A** veranschaulicht einen Ringoszillator, der ein AND-Gatter **231**, Inverter **233a-f**, Schalter **235a-c** und Konfigurationseingänge **237a-c** aufweist. In der veranschaulichenden Ausführungsform ist die PUF-Zelle **230** ein Ringoszillatorschaltkreis, der eine Ausgabe generiert, die zwischen einer logischen Eins und einer logischen Null oszilliert. Aufgrund von Prozessschwankungen kann es Unterschiede zwischen den beobachteten Oszillationsfrequenzen für Ringoszillatoren geben, die konstruiert sind, mit der gleichen Frequenz zu arbeiten.

[0041] Das AND-Gatter **231** kombiniert die Ausgabe des Ringoszillators mit einem Eingangssignal. Das Eingangssignal kann durch Steuerverschaltung (z. B. **123**) bereitgestellt werden. In der veranschaulichenden Ausführungsform deaktiviert eine Eingabe von null den Ringoszillator, da die Ausgabe des AND-Gatters **231** auf null gehalten wird und nicht oszilliert. In der veranschaulichenden Ausführungsform kann eine Eingabe von 1 verwendet werden, um den Ringoszillator zu aktivieren, um zum Beispiel die Oszillato-

ren im Wettlauf einzusetzen. In einigen Ausführungsformen können mehrere PUF-Zellen im Wesentlichen gleichzeitig aktiviert sein (z. B. unter Verwendung einer gemeinsam genutzten Signalleitung).

[0042] Inverter **235a-f** kehren eine empfangene Eingabe logisch um und bilden eine logische Schleife auf Grundlage der Konfiguration der Schalter **235a-c**. In einigen Ausführungsformen kann die PUF-Zelle **230** ein Ringoszillator sein, der eine ungerade Anzahl von Invertem ohne konfigurierbare Pfade (zum Beispiel wie in **Fig. 2B)** einsetzt.

[0043] Die Schalter **235a-c** können verwendet werden, um die PUF-Zelle **230** durch Einrichten von Signalpfaden für den Ringoszillator zu konfigurieren. In einigen Ausführungsformen können die Schalter **235a-c** eine oder mehrere Multiplexer oder ein beliebiges geeignetes Schaltelement sein.

[0044] In einigen Ausführungsformen kann die PUF-Zelle durch Anlegen von Konfigurationssignalen auf die Eingänge **237a-c** konfiguriert werden. In einigen Ausführungsformen können die Eingänge **237a-c** verwendet werden, um einen Signalpfad für den Ringoszillator auszuwählen. In der veranschaulichenden Ausführungsform von **Fig. 2A** können die Schalter **235a-c** und Eingänge **237a-c** verwendet werden, um einen von acht möglichen Pfaden durch die Inverter **233a-f** auszuwählen. Der Ringoszillator kann zum Beispiel nur einen der Inverter **233a-b**, einen der Inverter **233c-d** und einen der Inverter **233e-f** einsetzen. In einigen Ausführungsformen wird jede PUF-Zelle einmal konfiguriert, zum Beispiel während des Fertigungsprozesses. In einigen Ausführungsformen sind Konfigurationen ausgewählt, um die Zuverlässigkeit von schwachen PUF-Zellen zu erhöhen und zuverlässigere Wettlaufergebnisse zu erzeugen.

[0045] **Fig. 2B** zeigt ein veranschaulichendes Blockdiagramm einer PUF-Zelle **238**, die verwendet wird, um eine PUF-Ausgabe zu generieren, nach einigen Ausführungsformen. **Fig. 2B** veranschaulicht einen Ringoszillator, der ein AND-Gatter **239** und Inverter **233g**-i veranschaulicht. In der veranschaulichenden Ausführungsform von **Fig. 2B** ist die PUF-Zelle **238** ein Ringoszillatorschaltkreis, der eine Ausgabe generiert, die zwischen einer logischen Eins und einer logischen Null oszilliert. Aufgrund von Prozessschwankungen kann es Unterschiede zwischen den beobachteten Oszillationsfrequenzen für Ringoszillatoren geben, die konstruiert sind, mit der gleichen Frequenz zu arbeiten.

[0046] AND-Gatter **239** kombiniert die Ausgabe des Ringoszillators mit einem Eingangssignal. Das Eingangssignal kann durch Steuerverschaltung (z. B. **123**) bereitgestellt werden. In der veranschaulichenden Ausführungsform deaktiviert eine Eingabe von null den Ringoszillator, da die Ausgabe des AND-Gatters **239** auf null gehalten wird und nicht oszilliert. In der veranschaulichenden Ausführungsform kann eine Eingabe von 1 verwendet werden, um den Ringoszillator zu aktivieren, um zum Beispiel die Oszillatoren im Wettlauf einzusetzen. In einigen Ausführungsformen können mehrere PUF-Zellen im Wesentlichen gleichzeitig aktiviert sein (z. B. unter Verwendung einer gemeinsam genutzten Signalleitung).

[0047] Die Inverter **235a-f** kehren eine empfangene Eingabe logisch um. Die Ausgabe des Inverters **233i** wird in das AND-Gatter **239** rückgekoppelt, um eine fortlaufende Oszillation zu erzeugen. In einigen Ausführungsformen kann die Ausgabe des Inverters **233i** auf eine beliebige geeignete Weise gepuffert werden. In einigen Ausführungsformen wird die Ausgabe des Inverters **233i** einem Multiplexer, Zähler, Komparator und/oder einem beliebigen anderen geeigneten Schaltkreiselement zum Vergleichen von Ausgaben von PUF-Zellen (z. B. **230** und **238**) bereitgestellt.

[0048] **Fig. 3** zeigt ein veranschaulichendes Blockdiagramm eines PUF-Kerns **305** zum Generieren einer PUF-Ausgabe nach einigen Ausführungsformen. **Fig. 3** veranschaulicht eine Steuerverschaltung **341**, PUF-Zellen **330a-d**, Ausgabepuffer **342a-d**, Multiplexer **343a-b**, Zähler **345a-b** und einen Komparator **347**. In einigen Ausführungsformen kann der PUF-Kern **305** hinreichend viele Multiplexer, Zähler und Komparatoren aufweisen, um alle der PUF-Zellen **330a-d** zu paaren und jedes Paar parallel im Wettlauf einzusetzen. Beispielsweise durch Aufnehmen eines Multiplexers für jede PUF-Zelle **330a-d**, die in jeder Runde im Wettlauf einzusetzen ist.

[0049] In einigen Ausführungsformen ist der PUF-Kern **305** ausgebildet, ein oder mehrere Paare von PUF-Zellen iterativ im Wettlauf einzusetzen, zum Beispiel, um den Energieverbrauch des PUF-Kerns **305** zu reduzieren. In einigen Ausführungsformen sind die PUF-Zellen **330a-d** ausgebildet, die Wettläufe parallel zu beginnen. In weiteren Ausführungsformen wird die Ausgabe jeder PUF-Zelle gelatcht und/oder für einen nachfolgenden Vergleich zeitweilig gespeichert.

[0050] Die Steuerverschaltung **341** ermöglicht die Generierung einer Ausgabe von jeder der PUF-Zellen. In einigen Ausführungsformen generiert jede PUF-Zelle nur eine Ausgabe als Reaktion auf ein Signal von der Steuerverschaltung **341**. In einigen Ausführungsformen ordnet die Steuerverschaltung Paarungen von PUF-Zellen durch Steuern von PUF-Zelleneingaben zu. In einigen Ausführungsformen sind die Ausgänge der PUF-Zellen mit Multiplexern verbunden und die Steuerverschaltung **341** steuert, welche PUF-Zellen unter Verwendung der Multiplexer **343a-b** im Wettlauf eingesetzt werden.

[0051] Die PUF-Zellen **330a-d** generieren Ausgaben auf Grundlage der eindeutigen Hardwaremerkmale,

die für jeden PUF-Zellenschaltkreis spezifisch sind. Beispiele von PUF-Zellen weisen RO-PUFs, Arbiter-PUFs, andere verzögerungsbasierte PUFs oder eine beliebige PUF-Implementierung, die zum Vergleichen von Ausgabewerten verwendbar ist, auf. Es sollte klar sein, dass eine beliebige Eigenschaft, die für die Vorrichtung eindeutig ist, die deterministisch gemessen und verglichen werden kann, ohne vorab vorhergesagt zu werden, verwendet werden kann, um eine PUF-Ausgabe zu generieren.

[0052] Die Ausgaben jeder der PUF-Zellen **330a-d** werden durch einen jeweiligen der Ausgabepuffer **342a-d** gepuffert. Die Ausgabepuffer **342a-d** können eine beliebige geeignete Verschaltung zum Puffern der Ausgaben jeder der PUF-Zellen **330a-d** sein. Jeder der Multiplexer **343a-b** wird verwendet, um die Ausgabe einer der PUF-Zellen **330a-d** auszuwählen, um ein Paar von PUF-Zellen zu bilden. Die Steuerverschaltung **341** ordnet PUF-Zellenpaarungen zu und konfiguriert die Eingaben jedes Multiplexers **343a-b**, um das Paar der PUF-Zellen im Wettlauf einzusetzen. In einigen Ausführungsformen ist jeder der Multiplexer **343a-b** mit allen, einer Teilmenge oder einer beliebigen geeigneten Kombination der PUF-Zellen **330a-d** verbunden.

[0053] Die Zähler **345a-b** empfangen die Ausgabe der PUF-Zellen, die im Wettlauf eingesetzt werden, und messen eine vergleichbare Ausgabe. In einigen Ausführungsformen sind die PUF-Zellen Ringoszillatoren und die Zähler zählen Übergänge in der PUF-Zellenausgabe, um ein Maß der Frequenz zu ermitteln. In einigen Ausführungsformen kann der Zähler ein Analog-digital-Wandler oder ein anderes geeignetes Schaltkreiselement zum Generieren einer Ausgabe, die für einen Vergleich geeignet ist, sein. In einigen Ausführungsformen können die Zähler **345a-b** PUF-Zellenausgaben für einen nachfolgenden Vergleich latchen.

[0054] Der Komparator **347** empfängt und vergleicht die Ausgabe der Zähler **345a-b**. In einigen Ausführungsformen ist der Komparator ein Arbiter oder ein anderer geeigneter binärer Komparator zum Ermitteln eines Siegers und eines Verlierers von jedem Paar von PUF-Zellen. In einigen Ausführungsformen generiert der Komparator ein Ausgabebit. Das Ausgabebit kann ein volles Entropiebit repräsentieren, wie in Bezug auf PUF-Zellenpaarungen in **Fig. 1** besprochen wurde. Um eine geeignete Anzahl von Entropiebits, z. B. 32, 64, 128 oder 256 Entropiebits zu generieren, kann der PUF-Kern **305** die PUF-Zellen unter Verwendung von Paarungen, die von den Ergebnissen von vorangehenden Runden abhängen können, in mehreren Runden paaren und im Wettlauf einsetzen.

[0055] **Fig. 4A-C** zeigen veranschaulichende Runden einer Wettlaufreihenfolge für PUF-Zellen **430a-h** nach einigen Ausführungsformen. **Fig. 4A** repräsentiert eine veranschaulichende erste Runde von Paarungen **351a-d**. **Fig. 4B** repräsentiert eine veranschaulichende zweite Runde von Paarungen **353a-b** und **355a-b**. **Fig. 4C** repräsentiert eine veranschaulichende dritte Runde von Paarungen **357a-b** und **359a-b**. In der veranschaulichenden Ausführungsform der **Fig. 4A-C** werden acht PUF-Zellen **430a-h** im Wettlauf gegeneinander eingesetzt, und ein Einschränken der Anzahl der Runden auf den binären Logarithmus der Anzahl der PUF-Zellen, 3 Runden in den **Fig. 4A-C**, ermöglicht, dass durch jeden Wettlauf ein volles Entropiebit generiert wird. Zusätzlich werden nach einer Ausführungsform PUF-Zellen einmal pro Runde im Wettlauf eingesetzt. Das einmalige Einsetzen einer PUF-Zelle im Wettlauf pro Runde kann sicherstellen, dass jeder Wettlauf ein volles Entropiebit erzeugt. Im Beispiel von **Fig. 4A-C** verringert sich die Geschwindigkeit der PUF-Zellen **430a-h** in alphabetischer Reihenfolge, z. B. gewinnt die PUF-Zelle **430a** gegen die PUF-Zelle **430b**, die gegen PUF-Zelle **430c** gewinnt, die gegen PUF-Zelle **430d** gewinnt, und so weiter bis zur langsamsten PUF-Zelle **430h**, die keine Runden gewinnen wird. Die Reihenfolge der Geschwindigkeit der PUF-Zelle ist dem PUF-Kern vor den Wettläufen nicht bekannt und ist nur zur Illustration vorgesehen. Die PUF-Zellen **430a-h** können eine beliebige Reihenfolge der Geschwindigkeiten für jede PUF-Vorrichtung zeigen und jede Reihenfolge kann für die Vorrichtung einzigartig sein.

[0056] **Fig. 4A** zeigt eine veranschaulichende erste Runde von Paarungen und Wettläufen. In der ersten Runde ist jede PUF-Zelle mit einer benachbarten PUF-Zelle gepaart. PUF-Zelle **430a** ist mit PUF-Zelle **430b** gepaart. Jede PUF-Zelle im Paar generiert eine Ausgabe und die PUF-Zelle **430a** wird als der Sieger von Wettlauf **351a** ermittelt. PUF-Zelle **430c** ist mit PUF-Zelle **430d** gepaart und PUF-Zelle **430c** ist der Sieger des Wettlaufs **351b**. PUF-Zelle **430e** ist mit PUF-Zelle **430f** gepaart und PUF-Zelle **430e** ist der Sieger des Wettlaufs **351c**. PUF-Zelle **430g** ist mit PUF-Zelle **430h** gepaart und PUF-Zelle **430g** ist der Sieger des Wettlaufs **351d**. Die Wettläufe **351a-d** generieren jeweils ein volles Bit von Entropie. Die Paarungen für nachfolgende Runden werden auf Grundlage der Ergebnisse der ersten, in **Fig. 4A** gezeigten Runde ermittelt.

[0057] **Fig. 4B** zeigt eine veranschaulichende zweite Runde von Paarungen und Wettläufen. Im Beispiel von **Fig. 4B** sind die Wettläufe **353a-b** zwischen PUF-Zellen mit einem Sieg auf der rechten Seite von **Fig. 4B** gezeigt. Die Wettläufe **355a-b** sind zwischen PUF-Zellen mit einem Verlust und sind auf der linken Seite von **Fig. 4B** gezeigt.

[0058] Im Beispiel der **Fig. 4A-B** werden die Paarungen für die Wettläufe **353a-b** und **355a-b** durch Auswählen von zwei Paaren (z. B. aus den Wettläu-

fen **351a** und **351b)** aus der ersten Runde und Paaren der Sieger jedes ausgewählten Paars und der Verlierer jedes ausgewählten Paars ermittelt. In einigen Ausführungsformen werden die Paare sequenziell geordnet und die Reihenfolge der zweiten Runde wird durch Paaren des Wettlaufsiegers (Verlierers) von Paar i mit dem Sieger (Verlierer) von Paar i+1 ermittelt.

[0059] Im Beispiel von **Fig. 4B** werden die PUF-Zellen **430a** und **430c** gepaart, da jede die Wettläufe **351a** bzw. **351b** gewonnen hat. Entsprechend waren die PUF-Zellen **430b** und **430d** die Verlierer der Wettläufe **351a** und **351b** und werden in der zweiten Runde gepaart. Zusätzlich werden die PUF-Zellen **430e** und **430g** gepaart, da jede die Wettläufe **351c** bzw. **351d** gewonnen hat. Die PUF-Zellen **430f** und **430h** waren die Verlierer der Wettläufe **351c** und **351d** und werden in der zweiten Runde gepaart.

[0060] **Fig. 4C** zeigt eine veranschaulichende dritte Runde von Paarungen und Wettläufen. In der dritten Runde können die Sieger (Verlierer) der vorangehenden Runde in einigen Ausführungsformen mit einem Sieger (Verlierer) der vorangehenden Runde gepaart werden. In einigen Ausführungsformen werden die PUF-Zellen auf Grundlage davon gepaart, ob jede Zelle einen jeweiligen Wettlauf zwischen PUF-Zellen gewonnen oder verloren hat, die zwei Runden zuvor gewonnen oder verloren haben. Die PUF-Zellen **430b** und **430f** werden zum Beispiel im Wettlauf eingesetzt, nachdem sie die Verliererseite von **Fig. 4B** gewonnen haben.

[0061] In einigen Ausführungsformen werden PUF-Zellen mit identischen Verläufen gepaart. In weiteren Ausführungsformen traten die in der dritten Runde im Wettlauf eingesetzten PUF-Zellen weder in der ersten noch der zweiten Runde gegen eine der gleichen PUF-Zellen im Wettlauf an. In einigen Ausführungsformen werden die PUF-Zellen so gepaart, dass die PUF-Zellen im Paar ihre jeweiligen Wettläufe in der gleichen Reihenfolge gewonnen oder verloren haben. Die PUF-Zellen **430b** und **430f** werden zum Beispiel in Runde **3** gepaart, nachdem sie in Runde **1** verloren haben und in Runde **2** gewonnen haben.

[0062] In einigen Ausführungsformen werden die PUF-Zellen **430a-h** durch Kombinieren von Paaren von PUF-Zellen gruppiert. Die Gruppierung der PUF-Zellen kann virtuell unter Verwendung einer beliebigen geeigneten Datenstruktur oder Reihenfolge von Wettlaufsteuersignalen erfolgen. In einigen Ausführungsformen können Gruppen von PUF-Zellen sequenzielle und/oder zusammenhängende Abschnitte einer geordneten Liste von PUF-Zellen sein. In einigen Ausführungsformen kann eine neue Gruppe als eine Anzahl, zwei hoch der Anzahl der Runden (2^r2^r, wobei r die Anzahl der Runden ist), von zusammenhängenden PUF-Zellen aufweisend angesehen werden. In weiteren Ausführungsformen werden Gruppen von PUF-Zellen unter Verwendung einer oder mehrerer Hashtabellen, Speicherarrays oder Nachschlagetabellen nachverfolgt.

[0063] PUF-Zellen können so gruppiert werden, dass PUF-Zellen, die in einer vorangehenden Runde gepaart wurden, zusammen gruppiert sind. In einigen Ausführungsformen werden nachfolgende Paarungsrunden durch Auswählen von zwei Gruppen von PUF-Zellen und Paaren jeder PUF-Zelle in der ersten Gruppe mit einer jeweiligen PUF-Zelle in der zweiten Gruppe zugeordnet. Im Beispiel von **Fig. 4A** können die PUF-Zellen in jedem Wettlauf **351a-d** Gruppen mit jeweils zwei PUF-Zellen bilden. Im Beispiel von **Fig. 4B** werden die Gruppe mit den PUF-Zellen **430a-b** und die Gruppe mit den PUF-Zellen **430c-d** kombiniert, um eine Gruppe mit PUF-Zellen **430a-d** zu bilden. Eine zweite Gruppe kann gebildet werden, um die andere Hälfte der PUF-Zellen aufzunehmen, die PUF-Zellen **430e-h**. Die Gruppe, die die Hälfte der PUF-Zellen aufweisen, werden verwendet, um Paarungen für die dritte Runde zuzuordnen. Durch Auswählen von Paaren mit jeweils einer PUF-Zelle in einer der zwei Gruppen gibt es keine Überschneidung in den vorangehenden Wettlaufgegnern in beiden PUF-Zellen, da sich alle vorangehenden Wettlaufgegner einer PUF-Zelle in der gleichen Gruppe befinden. Deshalb hat der PUF-Kern keine A-priori-Informationen über das Ergebnis der Wettläufe und ein volles Entropiebit wird generiert.

[0064] In einigen Ausführungsformen kann eine Gruppierung von PUF-Zellen verwendet werden, um bestimmte Paare zuzuordnen. Zusätzlich zum Paaren von PUF-Zellen, die eine identische Anzahl von Paarungen gewonnen haben, kann es zum Beispiel wünschenswert sein, PUF-Zellen zu paaren, die Paarungen in der gleichen Reihenfolge gewonnen oder verloren haben. Im Beispiel des Paarens jeder PUF-Zelle in einer ersten Gruppe mit jeder PUF-Zelle in einer zweiten Gruppe, beginnend mit den Paaren in Runde **2**, da jede PUF-Zelle in einem Paar den gleichen Verlauf aufweist und eine PUF-Zelle gewinnt und eine PUF-Zelle verliert, weist jede PUF-Zelle eine Reihe von Siegen und Verlusten auf, die sich von den Siegen und Verlusten jeder anderen PUF-Zelle in der Gruppe in mindestens einer Runde unterscheidet. Deshalb weist jede PUF-Zelle in der Gruppe eine eindeutige Reihe von Siegen und Verlusten auf. Darüber hinaus, da die Anzahl der PUF-Zellen in jeder Gruppe gleich der Anzahl der möglichen Reihenfolgen von Siegen und Verlusten ist, zwei hoch der Anzahl der Runden, weist jede Gruppe eine PUF-Zelle für jede mögliche eindeutige Reihe von Siegen und Verlusten auf. Paarungen für nachfolgende Runden können deshalb durch Abstimmen von PUF-Zellen mit identischen Verläufen erfolgen, da die auf die beispielhafte Weise gepaarten Gruppen eine identische Größe aufweisen. Gruppen von PUF-Zellen können

auf beliebige geeignete Weise kombiniert oder gepaart werden. In einigen Ausführungsformen kann eine Reihe von Siegen und Verlusten unter Verwendung einer Nachschlagetabelle, einer Anordnung, einer Bitmap oder einer beliebigen anderen geeigneten Datenstruktur nachverfolgt werden.

[0065] Wie oben besprochen, nachdem die Anzahl der Runden den binären Logarithmus der Anzahl der PUF-Zellen überschreitet ($\log_2(N)$, wobei N die Anzahl der PUF-Zellen ist) kann die Entropie jeder Paarung geringer als ein volles Bit sein. In einigen Ausführungsformen kann der PUF-Kern mit zusätzlichen Runden fortfahren, um die Wettlaufreihenfolge der PUF-Zellen weiter zu ermitteln. Der PUF-Kern kann zum Beispiel PUF-Zellen im Wettlauf einsetzen, die in der vorangehenden Runde Sieger (Verlierer) gegen einen anderen Sieger (Verlierer) der vorangehenden Runde waren, falls die zwei gepaarten Zellen nicht vorher gepaart wurden. In einigen Fällen können diese zusätzlichen Wettläufe abhängig von der Topologie der PUF-Zellen null Informationen ergeben. Es kann zum Beispiel aus der transitiven Eigenschaft offensichtlich sein, dass eine PUF-Zelle gegen eine gepaarte PUF-Zelle gewinnen wird, falls die PUF-Zellen gegen den gleichen Gegner gewonnen bzw. verloren haben.

[0066] **Fig. 5A** zeigt ein veranschaulichendes Zeitgebungsdiagramm für Wettläufe von PUF-Zellen und zum Generieren von fehlerkorrigierten Ergebnissen, nach einigen Ausführungsformen. **Fig. 5A** veranschaulicht einen Prozessablauf **500**, der unter Verwendung einer Vielzahl von PUF-Zellen, zum Beispiel von einem PUF-Kern (z. B. **105**) ausgeführt werden kann.

[0067] Der Prozessablauf **500** beginnt bei Handlung **501**, wobei N PUF-Zellen in einigen Ausführungsformen in N/2 Paaren im Wettlauf eingesetzt werden. In einigen Ausführungsformen können die Wettläufe in Handlung **501** iterativ ausgeführt werden und nach einer bestimmten Anzahl von Iterationen kann eine Decodierung durchgeführt werden. Die N/2 Paare von PUF-Zellen können zum Beispiel wie unter Bezugnahme auf **Fig. 1**, **Fig. 3**, **Fig. 5**, **Fig. 8** und **Fig. 9** beschrieben im Wettlauf eingesetzt werden.

[0068] Bei Handlung **503** werden die Ergebnisse der N/2 Wettläufe, N/2 Bits, unter Verwendung eines beliebigen geeigneten Fehlerkorrekturcodes (z. B. dem ECC-Decodierer **119**) decodiert. In einigen Ausführungsformen können die Ausgaben einer beliebigen PUF-Zelle verrauscht sein. In einigen Ausführungsformen können die PUF-Zellen ausgebildet sein, einen robusten Vergleich zu erzeugen, zum Beispiel Wettläufe, die nicht von der Antworttemperatur jeder PUF-Zelle beeinflusst sind, aber schwache PUF-Zellen oder enge, verrauschte Wettläufe sind weiterhin möglich. In einigen Ausführungsformen kann die Fehlerkorrektur unter Verwendung eines ECC-Decodierers (z.B. **119**) durchgeführt werden, der mit verschiedenen Abschnitten einer Kryptografievorrichtung (z. B. **100**) gemeinsam genutzt wird. In einigen Ausführungsformen werden ECC-Daten während der Verschlüsselungs-/Festschreibphase für jede Runde von Wettlaufergebnissen erstellt.

[0069] Bei Handlung **505** wird eine zweite Runde von Wettläufen unter Verwendung von Paarungen auf Grundlage der Ergebnisse der ersten Runde durchgeführt. In der veranschaulichenden Ausführungsform von **Fig. 5A** wird jede Paarung in Handlung **505** erst ermittelt, nachdem die zur Ermittlung der Paarung notwendigen Ergebnisse in Handlung **503** decodiert wurden. In einigen Ausführungsformen werden alle Ergebnisse der ersten Runde gleichzeitig im gleichen Fehlerkorrekturprozess decodiert. In einigen Ausführungsformen beginnt die Zuordnung der Paare der zweiten Runde erst, wenn alle Ergebnisse der ersten Runde decodiert wurden. In einigen Ausführungsformen werden Wettläufe (z. B. 8, 16, 32 oder 64 Wettläufe) decodiert und eine zweite Runde von Paarungen kann auf decodierten Abschnitten der ersten Runde von Ergebnissen ausgewählt werden. In einigen Ausführungsformen fährt die zweite Wettlaufrunde spekulativ fort, sobald die Wettläufe in Handlung **501** abgeschlossen sind. Die Ergebnisse von spekulativen Wettläufen können als gültige Ergebnisse bestätigt werden, falls keine Fehler in Handlung **503** erkannt werden. Falls ein Fehler erkannt wird, können die Ergebnisse von spekulativen Wettläufen verworfen und/oder ungültig gemacht werden und zumindest die vom Fehler betroffenen Wettläufe können wiederholt werden.

[0070] Der Prozessablauf **500** endet bei Handlung **507**, wo die Ergebnisse der zweiten Wettlaufrunde unter Verwendung eines Fehlerkorrekturcodes ähnlich wie bei Handlung **503** decodiert werden. In einigen Ausführungsformen werden die decodierten Ergebnisse als Ausgabe, eine PUF-Ausgabe, bereitgestellt und N Entropiebits aus N Wettläufen und N PUF-Zellen sind ausreichend. In einigen Ausführungsformen kann die Anzahl der PUF-Zellen kleiner als die gewünschte Anzahl von Entropiebits sein und der Prozessablauf fährt auf eine ähnliche serialisierte Weise mit zusätzlichen Runden fort.

[0071] **Fig. 5B** zeigt ein veranschaulichendes Zeitgebungsdiagramm für Wettläufe von PUF-Zellen und zum Generieren von fehlerkorrigierten Ergebnissen, nach einigen Ausführungsformen. **Fig. 5B** veranschaulicht einen Prozessablauf **510**, der unter Verwendung einer Vielzahl von PUF-Zellen, zum Beispiel von einem PUF-Kern (z. B. **105**) ausgeführt werden kann.

[0072] Der Prozessablauf **510** beginnt bei Handlung **511**, wobei N PUF-Zellen in einigen Ausführungsfor-

men in N/2 Paaren im Wettlauf eingesetzt werden. Die N/2 Paare von PUF-Zellen können zum Beispiel wie unter Bezugnahme auf Handlung **501** und **Fig. 1**, **Fig. 3**, **Fig. 5**, **Fig. 8** und **Fig. 9** beschrieben im Wettlauf eingesetzt werden.

[0073] Bei Handlung **513** werden die Ergebnisse der N/2 Wettläufe, N/2 Bits, unter Verwendung eines beliebigen geeigneten Fehlerkorrekturcodes (z. B. dem ECC-Decodierer **119**) decodiert werden. In einigen Ausführungsformen können die Ausgaben einer beliebigen PUF-Zelle verrauscht sein. In einigen Ausführungsformen können die PUF-Zellen ausgebildet sein, einen robusten Vergleich zu erzeugen, zum Beispiel Wettläufe, die nicht von der Antworttemperatur jeder PUF-Zelle beeinflusst sind, aber schwache PUF-Zellen oder enge, verrauschte Wettläufe sind weiterhin möglich. In einigen Ausführungsformen kann die Fehlerkorrektur unter Verwendung eines ECC-Decodierers (z.B. **119**) durchgeführt werden, der mit verschiedenen Abschnitten einer Kryptografievorrichtung (z. B. **100**) gemeinsam genutzt wird.

[0074] In einigen Ausführungsformen werden die decodierten Ergebnisse im endgültigen PUF-Ausgabewert verwendet, werden jedoch nicht für die restlichen Handlungen des Prozessablaufs **510** benötigt und können zu einem beliebigen geeigneten Zeitpunkt decodiert werden. In einigen Ausführungsformen kann die Decodierung in den Handlungen **513** und **517** im Wesentlichen gleichzeitig durchgeführt werden.

[0075] Bei Handlung **515** wird eine zweite Runde von Wettläufen unter Verwendung von Paarungen auf Grundlage der Ergebnisse der ersten Runde durchgeführt. In der veranschaulichenden Ausführungsform von **Fig. 5B** wird jede Paarung in Handlung **515** ohne die fehlerkorrigierten Ergebnisse der Wettläufe in Handlung **511** ermittelt. Da die Paare auf Grundlage der Ergebnisse der vorangehenden Runde zugeordnet werden, kann ein Bitfehler in der Ausgabe zu zwei fehlerhaften Paarungen in der zweiten Runde und deshalb zwei Fehlern in den Ergebnissen der zweiten Runde führen. In einigen Ausführungsformen können sich die Fehler so fortpflanzen, dass sich die Anzahl der Bitfehler in jeder Runde verdoppelt. Ein Korrigieren dieser Fehlerfortpflanzung kann einen wesentlich komplexeren Fehlerkorrekturcode erfordern, als für den serialisierten Prozessablauf **500** erforderlich ist. Es kann zum Beispiel notwendig sein, dass die Fehlerkorrektur im Prozess **510** an allen Ergebnissen gleichzeitig operiert und/oder mehr als doppelt so viele Fehler erkennt/korrigiert.

[0076] Der Prozessablauf **510** endet bei Handlung **517**, wo die Ergebnisse der zweiten Wettlaufrunde unter Verwendung eines Fehlerkorrekturcodes ähnlich wie bei Handlung **513** decodiert werden. In einigen Ausführungsformen werden die decodierten Ergebnisse als Ausgabe, eine PUF-Ausgabe, bereitgestellt und N Entropiebits aus N Wettläufen und N PUF-Zellen sind ausreichend. In einigen Ausführungsformen kann die Anzahl der PUF-Zellen kleiner als die gewünschte Anzahl von Entropiebits sein und der Prozessablauf fährt mit zusätzlichen Runden fort.

[0077] **Fig. 6** zeigt einen veranschaulichenden Prozessablauf **600** für eine Festschreibungsphase zum Codieren eines Geheimnisses unter Verwendung einer PUF nach einigen Ausführungsformen. Der in **Fig. 6** veranschaulichte Prozess kann durch eine Kryptografievorrichtung (z. B. **100**) ausgeführt werden, die ausgebildet ist, eine PUF-Ausgabe (z. B. **107** und **115**) zu generieren. In einigen Ausführungsformen bindet der Prozess **600** ein Geheimnis an unter Verwendung der PUF generierte Hilfsdaten oder bildet ein Geheimnis auf diese ab.

[0078] Der beispielhafte Prozessablauf **600** beginnt bei Handlung **601**, wobei die Verschlüsselungsvorrichtung ein Geheimnis K bezieht. Das Geheimnis kann eine beliebige Reihe von Bits sein, die zur Eingabe in die Verschlüsselungsvorrichtung geeignet sind, die der Benutzer der Vorrichtung geheim halten will, zum Beispiel ein privater Verschlüsselungsschlüssel. Das Geheimnis wird der Vorrichtung als Eingabe bereitgestellt und in einigen Ausführungsformen können mehrere Geheimnisse unter Verwendung der Vorrichtung eindeutig verschlüsselt und entschlüsselt werden.

[0079] Bei Handlung **603** verwendet die Verschlüsselungsvorrichtung einen ECC-Codierer (z. B. **103**), um einen ECC auf das Geheimnis K anzuwenden, um ECC-Daten zu erzeugen. Bei Handlung **605** werden die vom ECC-Codierer generierten ECC-Daten an das Geheimnis K angefügt. Die ECC-Daten können auch in einem beliebigen geeigneten Speichermedium gespeichert werden und können mit dem Geheimnis assoziiert werden, ohne direkt angefügt zu werden. Der ECC-Codierer kann ausgebildet sein, unter Verwendung eines beliebigen geeigneten ECC-Codes ECC-Daten zu berechnen und eine beliebige Anzahl von Bitfehlern zum erfolgreichen Wiederherstellen des Geheimnisses korrigieren. In einigen Ausführungsformen führt der ECC-Codierer eine Bose-Chaudhuri-Hocquenghem(BCH)-Codierung durch.

[0080] Bei Handlung **607** generiert die Verschlüsselungsvorrichtung unter Verwendung eines PUF-Kerns (z. B. des PUF-Kerns **105**) eine PUF (z. B. die PUF **107**). Der PUF-Kern kann ein beliebiger geeigneter Schaltkreis oder ein beliebiges geeignetes System zum Generieren einer PUF sein. In einigen Ausführungsformen kann der PUF-Kern Verschaltung aufweisen, die eine Ausgabe generiert, die von eindeutigen physikalischen Eigenschaften von einer oder mehreren PUF-Zellen abhängt, die im PUF-Kern enthalten sein können. Schwankungen in Fertigungs-

prozessen und in Teilen können zum Beispiel einen Chip erzeugen, der elektrische Schaltkreise mit eindeutigen Hardwaremerkmalen aufweist. Beispiele von PUF-Zellen weisen Ringoszillatoren, Arbiter-PUFs, andere verzögerungsbasierte PUFs oder eine beliebige PUF-Implementierung, die zum Vergleichen von Ausgabewerten von identisch konstruierten Schaltkreisen verwendbar sind, auf. In einigen Ausführungsformen wird der PUF-Kern zum Beispiel von der Steuerverschaltung **123** angewiesen, eine PUF-Ausgabe (z.B. **107** oder **115**) zu generieren und erzeugt eine PUF-Ausgabe, die nicht auf irgendeinem Eingabewert in den PUF-Kern basiert.

[0081] In einigen Ausführungsformen ist die PUF eine Reihe von Bits, die zum Verschleiern des Geheimnisses **101** geeignet ist. In einigen Ausführungsformen weist jede PUF eine Anzahl von Bits (z. B. 32, 64, 128 oder 256 Bits) von Entropie auf, die durch Wettlaufen von Paaren von PUF-Zellen generiert wurden, wobei das Ergebnis jedes Wettlaufs einer jeweiligen Bitposition in der PUF entspricht. In einigen Ausführungsformen repräsentiert jedes Bit in der PUF ein volles Entropiebit. In einigen Ausführungsformen werden ECC-Daten zum Decodieren eines zukünftigen PUF-Resultats während der Festschreibphase und Handlung **607** generiert. In der gesamten Offenbarung werden Systeme und Verfahren zum Generieren von PUFs beschrieben, z. B. unter Verwendung einer optimalen Wettlaufstrategie von PUF-Zellen.

[0082] Bei Handlung **609** führt die Verschlüsselungsvorrichtung eine Operation (z. B. unter Verwendung von Operator **109**) durch, um unter Verwendung von PUF-Daten das Geheimnis K und die angefügten ECC-Daten zu verschleiern, um ein Ausgabewort W (z.B. Hilfsdaten **111**) zu erzeugen. Die Operation kann eine beliebige geeignete Operation zur Verwendung der PUF sein, um das Geheimnis K zu verschleiern, sodass die Hilfsdaten W nicht verwendet werden können, um das Geheimnis K ohne die korrekte PUF zu ermitteln. Die Operation kann eine beliebige umkehrbare Operation oder Reihe von Operationen sein, um eine Wiederherstellung des Geheimnisses K aus Hilfsdaten W zu ermöglichen. In einigen Ausführungsformen ist die Operation eine bitweise XOR-Operation, die ihr eigenes Inverses sein kann. In einigen Ausführungsformen maskiert die Operation das Geheimnis K durch Ändern der Ausgabebits, die Bits im Geheimnis K entsprechen, auf Grundlage der PUF.

[0083] **Fig. 7** zeigt einen veranschaulichenden Prozessablauf **700** für eine Wiederherstellungsphase zum Decodieren eines Geheimnisses unter Verwendung einer PUF nach einigen Ausführungsformen. Der in **Fig. 7** veranschaulichte Prozess kann durch eine Kryptografievorrichtung (z. B. **100**) ausgeführt werden, die ausgebildet ist, eine PUF-Ausgabe (z. B. **107** und **115**) zu generieren. In einigen Ausführungsformen generiert der Prozess **700** ein codiertes Geheimnis unter Verwendung von während der Festschreibphase generierten Hilfsdaten erneut.

[0084] Der beispielhafte Prozessablauf **700** beginnt bei Handlung **701**, wobei die Verschlüsselungsvorrichtung ein Eingabewort (z. B. **113**) bezieht. Das Eingabewort kann eine beliebige Reihe von Bits sein, die für die Eingabe in die Verschlüsselungsvorrichtung geeignet ist. Das Eingabewort kann verwendet werden, um das Geheimnis zu entschlüsseln, falls die Eingabehilfsdaten mit den Ausgabehilfsdaten vom Verschlüsselungsprozess übereinstimmen.

[0085] Bei Handlung **703** generiert die Verschlüsselungsvorrichtung unter Verwendung eines PUF-Kerns (z. B. des PUF-Kerns **105**) eine PUF-Ausgabe (z. B. **115**). Die PUF-Ausgabe kann wie unter Bezugnahme auf **Fig. 3**, **Fig. 4A-C** und **Fig. 5A-B** besprochen generiert werden. In einigen Ausführungsformen wird die Ausgabe des PUF-Kerns in Runden generiert, wobei die zugeordneten Paarungen nach der ersten Runde von den Ergebnissen der vorangehenden Runden abhängen. In einigen Ausführungsformen wird die Generierung einer PUF-Ausgabe durch die Steuerverschaltung (z. B. **123**) oder einen Prozessor oder Steuerverschaltung im PUF-Kern gesteuert.

[0086] Bei Handlung **705** führt die Verschlüsselungsvorrichtung die Umkehrung (z. B. unter Verwendung des Operators **117**) einer Operation durch, die verwendet wurde (z. B. unter Verwendung des Operators **109**), um das Geheimnis K und die angefügten ECC-Daten während der Festschreibphase zu verschleiern, um das Geheimnis zu schätzen. Die Operation kann eine beliebige geeignete Operation oder eine beliebige geeignete Reihe von Operationen zum Umkehren der Operation(en) sein, die zum Verschleiern des Geheimnisses während der Verschlüsselung und zum Schätzen des Geheimnisses aus den Eingabehilfsdaten verwendet wurde(n). In einigen Ausführungsformen kann die Operation eine XOR- oder Maskierungsoperation sein.

[0087] Bei Handlung **707** verwendet die Verschlüsselungsvorrichtung den ECC, um die Schätzung des Geheimnisses (z. B. unter Verwendung des ECC-Decodierers **119**) zu decodieren, um ein wiederhergestelltes Geheimnis K zu erzeugen. Das ursprüngliche verschlüsselte Geheimnis kann aus den Eingabehilfsdaten **113** wiederhergestellt werden und stimmt mit dem Geheimnis überein, falls die Verschlüsselungsausgabehilfsdaten mit den Entschlüsselungseingabehilfsdaten übereinstimmen. In einigen Ausführungsformen werden die Verschlüsselungsausgabehilfsdaten im Arbeitsspeicher gespeichert oder an einen Benutzer gesendet. In einigen Ausführungsformen wird die Ausgabe des PUF-Kerns nicht gespei-

chert oder außerhalb der Verschlüsselungsvorrichtung gesendet.

[0088] **Fig. 8** zeigt einen veranschaulichenden Prozessablauf für Wettlaufrunden von PUF-Zellen, um eine PUF-Ausgabe zu generieren, nach einigen Ausführungsformen. Der in **Fig. 8** veranschaulichte Prozess kann von einer Kryptografievorrichtung (z. B. **100**) ausgeführt werden, die ausgebildet ist, eine PUF-Ausgabe (z.B. **107** und **115**) zu generieren, oder beliebige der hierin besprochenen Systeme und Verfahren, zum Beispiel die unter Bezugnahme auf **Fig. 1**, **Fig. 3**, **Fig. 4A-C** und **Fig. 5A** besprochenen.

[0089] Der Prozess **800** beginnt bei Handlung **801**, wobei die Verschlüsselungsvorrichtung eine Paarung der ersten Runde von PUF-Zellen zuordnet. In einigen Ausführungsformen kann die erste Paarung eine beliebige geeignete Abbildung von Paaren sein, die zum Reproduzieren der PUF-Ausgabe reproduziert werden kann. Zum Beispiel kann die erste Paarungsrunde benachbarte PUF-Zellen paaren. In einigen Ausführungsformen sind die PUF-Zellen geordnet und/oder von 1 bis N nummeriert. In der ersten Runde können die ungerade nummerierten PUF-Zellen mit einer benachbarten gerade nummerierten PUF-Zelle gepaart werden, wobei z. B. die PUF-Zellen **1** und **2**, die PUF-Zellen **3** und **4**, die PUF-Zellen **5** und **6**, ... und die PUF-Zellen N-1 und N gepaart werden. In einigen Ausführungsformen werden die PUF-Zellen mit einer verfügbaren PUF-Zelle gepaart, die um N/2 Plätze oder eine andere geeignete Distanz getrennt ist. In einigen Ausführungsformen repräsentiert die erste Runde von PUF-Zellen-Paarungen eine beliebige paarweise Abbildung von PUF-Zellen.

[0090] Bei Handlung **803** werden die in Handlung **801** gepaarten PUF-Zellen im Wettlauf eingesetzt, um einen Sieger und Verlierer jedes Paars für die Runde zu ermitteln. In einigen Ausführungsformen wird jedes Paar von PUF-Zellen in der ersten Paarung im Wettlauf gegeneinander eingesetzt, indem bewirkt und/oder ermöglicht wird, dass jede PUF-Zelle im Paar eine Ausgabe generiert. In einigen Ausführungsformen wird eine PUF-Zelle jedes Paars auf Grundlage eines binären Vergleichs der Ausgaben als der Sieger ermittelt und eine PUF-Zelle jedes Paars wird als der Verlierer ermittelt. Wenn die PUF-Zellen zum Beispiel Ringoszillatoren aufweisen, können Ausgabeübergänge jedes Oszillators für eine Zeitperiode gezählt und verglichen werden, wobei der Oszillator mit der höheren Frequenz einen größeren Zählerstand erzeugt und als der Sieger ermittelt wird. In einigen Ausführungsformen kann eine schiedsrichterbasierte PUF eine Schiedsrichterausgabe mit jedem von zwei Verzögerungspfaden assoziieren. Es sollte klar sein, dass andere verzögerungsbasierte PUFs als geeignete PUF-Zellen verwendet werden können.

[0091] Bei Handlung **805** werden die Ergebnisse der in Handlung **803** durchgeführten Wettlaufrunde unter Verwendung eines Fehlerkorrekturcodes decodiert. Die Ergebnisse können zum Beispiel wie unter Bezugnahme auf **Fig. 5A** besprochen decodiert werden. Die Ergebnisse der Wettläufe werden unter Verwendung eines beliebigen geeigneten Fehlerkorrekturcodes (z.B. dem ECC-Decodierer **119**) decodiert. In einigen Ausführungsformen können die Ausgaben einer beliebigen PUF-Zelle verrauscht sein. In einigen Ausführungsformen kann die Fehlerkorrektur unter Verwendung eines ECC-Decodierers (z. B. **119**) durchgeführt werden, der mit verschiedenen Abschnitten einer Kryptografievorrichtung (z. B. **100**) gemeinsam genutzt wird. In einigen Ausführungsformen werden ECC-Daten während der Verschlüsselungs-/Festschreibphase für jede Runde von Wettlaufergebnissen erstellt. In einigen Ausführungsformen können die ECC-Daten, die verwendet werden, um eine Runde von Wettlaufergebnissen zu decodieren, auf Grundlage eines Abschnitts eines codierten Geheimnisses (z. B. einer Anzahl von Bits, die gleich der Anzahl der Wettläufe in einer Runde ist) erstellt werden. In weiteren Ausführungsformen können die Ergebnisse der Runde unter Verwendung eines Abschnitts (z. B. einer Anzahl von Bits, die gleich der Anzahl von Wettläufen in einer Runde ist) eines wiederhergestellten Geheimnisses, das aus den Ergebnissen der Runde der Wettläufe und zum Codieren eines Geheimnisses verwendeten Hilfsdaten generiert wurde, decodiert werden.

[0092] Bei Handlung **807** prüft die Verschlüsselungsvorrichtung, ob eine Grenze der Anzahl der Runden erreicht wurde. Aus der vorangehenden Besprechung sollte klar sein, dass, um zu garantieren, dass jeder Wettlauf ein volles Entropiebit erzeugt, die Anzahl der Paarungsrunden auf den binären Logarithmus der Anzahl der PUF-Zellen, N, eingeschränkt werden kann. Nach diesem Punkt kann die Wettlaufreihenfolge der PUF-Zellen großteils ermittelt sein und zukünftige Wettläufe können weniger als ein volles Entropiebit ergeben. Nach N-1 Runden kann jede mögliche Paarung von PUF-Zellen im Wettlauf eingesetzt worden sein.

[0093] Falls die Rundengrenze noch nicht erreicht wurde, wiederholt sich der Prozess durch Zuordnen einer zusätzlichen Runde von Paarungen zu den PUF-Zellen auf Grundlage der Ergebnisse der vorangehenden Runde bei Handlung **809**.

[0094] Wie oben besprochen, wenn PUF-Zellen im Wettlauf gegeneinander eingesetzt werden, die vorher noch nicht miteinander und/oder einem gemeinsamen Gegner gepaart worden sind, kann ein volles Entropiebit durch das Resultat des Wettlaufs generiert werden, da es keine A-priori-Informationen gibt, die verwendet werden können, um das Resultat vorherzusagen. In einigen Ausführungsformen werden

durch Einsetzen in Wettläufen von Siegern (Verlierern) gegen Sieger (Verlierer) von vorangehenden Runden PUF-Zellen mit ähnlichen Verläufen und/oder identischen Verläufen an Siegen und Verlusten gepaart und dies kann ein volles Entropiebit erzeugen. In einigen Ausführungsformen werden N PUF-Zellen in N/2 Paaren pro Runde im Wettlauf eingesetzt. Deshalb können in jeder Runde, in der N/2 PUF-Zellen ohne gemeinsame Gegner gepaart werden, N/2 Entropiebits generiert werden. In zwei Runden können N PUF-Zellen N Entropiebits generieren.

[0095] In einigen Ausführungsformen werden die Paarungen der PUF-Zellen durch den PUF-Kern **105** und/oder die Steuerverschaltung **123** ausgewählt, um sicherzustellen, dass gepaarte PUF-Zellen keinen gemeinsamen vergangenen Gegner aufweisen. In einigen Ausführungsformen wird die Auswahl aktiv durchgeführt, zum Beispiel durch Nachverfolgung des Satzes von Gegnern, mit dem jede PUF-Zelle konfrontiert wird, und Auswählen von Paaren, deren Sätze von Gegnern einen Schnitt von null aufweisen. In einigen Ausführungsformen gruppiert die Wettlaufreihenfolge die PUF-Zellen, um gemeinsame vergangene Gegner zu verhindern.

[0096] In einigen Ausführungsformen wird die Wettlaufreihenfolge verwendet, um Bedingungen dafür zu schaffen, dass durch jede Paarung und/oder jeden Wettlauf eine volle Entropie generiert wird. In der ersten Runde können PUF-Zellen keinesfalls gemeinsame Gegner aufweisen, da keine Wettläufe durchgeführt worden sind. In einigen Ausführungsformen wird jeder Sieger in der ersten Runde mit einem anderen Sieger von der ersten Runde gepaart und jeder Verlierer wird mit einem anderen Verlierer gepaart. In einer zweiten Runde, die auf diese Weise zugeordnet wird, kann kein PUF-Zellenpaar einen überschneidenden Verlauf von Gegnern aufweisen, da jeder Sieger (Verlierer) vorher genau gegen einen jeweiligen Verlierer (Sieger) im Wettlauf angetreten ist. In einigen Ausführungsformen wird eine dritte Runde von Paarungen so zugeordnet, dass Sieger (Verlierer) der vorangehenden Runde mit Siegern (Verlierern) der vorangehenden Runde gepaart werden. In einigen Ausführungsformen werden PUF-Zellen gepaart, die den gleichen Verlauf aufweisen und die Wettläufe in der gleichen Reihenfolge gewonnen oder verloren haben.

[0097] Sobald die Rundengrenze erreicht ist, werden die Ergebnisse der Wettläufe bei Handlung **811** ausgegeben. In einigen Ausführungsformen nimmt das binäre Ergebnis jeder PUF-Zelle eine jeweilige Position in der Ausgabe ein. Eine erste Runde von Wettläufen kann zum Beispiel erste 64 Ausgabebits ergeben und eine zweite Runde von Wettläufen kann zweite 64 Ausgabebits ergeben. In einigen Ausführungsformen wird die Ausgabe jedes Wettlaufs auf eine Position in der PUF-Ausgabe in Übereinstimmung mit einer beliebigen geeigneten deterministischen Abbildung abgebildet.

[0098] **Fig. 9** zeigt einen veranschaulichenden Prozessablauf für ein Verfahren zum Generieren von Entropie in einer PUF-Codierung nach einigen Ausführungsformen. Der in **Fig. 9** veranschaulichte Prozess kann von einem beliebigen geeigneten Prozessor oder einer beliebigen geeigneten Steuerverschaltung (z. B. **105** oder **123**) ausgeführt werden, der bzw. die ausgebildet ist, eine PUF-Ausgabe (z. B. **107** und **115**) von beliebigen der hierin besprochenen Systeme und Verfahren zu generieren, zum Beispiel der unter Bezugnahme auf **Fig. 1**, **Fig. 3**, **Fig. 4A-C**, **Fig. 5A-B** und **Fig. 8** besprochenen.

[0099] Ein Prozess **900** beginnt mit Handlung **901**, wobei mindestens ein Prozessor in einer ersten Runde erste Paarungen von jeweiligen aus einer Vielzahl von PUF-Zellen zuordnet. In einigen Ausführungsformen weist jede PUF-Zelle einen jeweiligen Ringoszillatorschaltkreis auf. In einigen Ausführungsformen weist jede PUF-Zelle einen jeweils identisch konstruierten Schaltkreis auf, wobei Unterschiede zwischen den jeweiligen identisch konstruierten Schaltkreisen aus Schwankungen im Fertigungsprozess herrühren. Die Paarungen können zum Beispiel wie unter Bezugnahme auf eine der **Fig. 1**, **Fig. 4A** und **Fig. 8** besprochen zugeordnet werden.

[0100] Bei Handlung **903** werden in der ersten Runde jeweilige erste Ausgaben von jeder der Vielzahl der PUF-Zellen generiert. Die ersten Ausgaben werden verwendet, um die Ergebnisse für die erste Runde zu ermitteln, einschließlich eines Siegers für jedes Paar von PUF-Zellen in der ersten Paarung. In einigen Ausführungsformen kann Verarbeitungsverschaltung PUF-Zellen aktivieren und auswählen, die in den Wettläufen verwendet werden, wie zum Beispiel unter Bezugnahme auf **Fig. 3** besprochen wurde. In einigen Ausführungsformen werden Zähler und ein Komparator verwendet, um einen siegenden und verlierenden Ringoszillator zu ermitteln. In einigen Ausführungsformen ermittelt ein Arbiter einen siegenden und verlierenden Verzögerungspfad. In einigen Ausführungsformen ist der Sieger jedes Paars von PUF-Zellen mit einer Ausgabe eines binären Vergleichs jeweiliger Ausgaben jeder PUF-Zelle im Paar assoziiert.

[0101] In einigen Ausführungsformen wendet die Verarbeitungsverschaltung vor der zweiten Runde und Handlung **905** einen Fehlerkorrekturcode auf die jeweiligen ersten Ausgaben von jeder der Vielzahl der PUF-Zellen an. Die Wettlaufergebnisse können zum Beispiel wie unter Bezugnahme auf **Fig. 5A** besprochen decodiert werden. In einigen Ausführungsformen werden während einer Festschreibphase ECC-Hilfsdaten für die Wettlaufergebnisse berechnet, in der ein Geheimnis verschlüsselt wird.

[0102] Bei Handlung **905** werden zweite Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer zweiten, auf die erste Runde folgenden Runde zugeordnet. In einigen Ausführungsformen wird die zweite Paarungsrunde wie unter Bezugnahme auf **Fig. 1**, **Fig. 3**, **Fig. 5B** und **Fig. 8** besprochen zugeordnet. In einigen Ausführungsformen werden die Paarungen so zugeordnet, dass die jeweiligen Sieger jedes Paars von PUF-Zellen in der ersten Paarung mit einem anderen Sieger gepaart werden und verlierende PUF-Zellen mit verlierenden PUF-Zellen gepaart werden. In einigen Ausführungsformen wird ein erster Sieger eines ersten Paars in einer vorangehenden Runde mit einem zweiten Sieger eines zweiten Paars in der vorangehenden Runde gepaart. In einigen Ausführungsformen werden die PUF-Zellen in einem ersten Paar in der ersten Runde mit den PUF-Zellen in einem zweiten Paar in der ersten Runde gepaart. In einigen Ausführungsformen werden PUF-Zellen gepaart, die in der gleichen Reihenfolge Wettläufe gewonnen oder verloren haben. In einigen Ausführungsformen werden PUF-Zellen so gruppiert, dass PUF-Zellen, die am Wettlauf teilgenommen haben, in der gleichen Gruppe sind, und Paare können durch Paaren jeder PUF-Zelle in einer ersten Gruppe mit einer jeweiligen PUF-Zelle in einer zweiten Gruppe zugeordnet werden.

[0103] Bei Handlung **907** werden jeweilige zweite Ausgaben von jeder der Vielzahl von PUF-Zellen in der zweiten Runde generiert und Ergebnisse für die zweite Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den zweiten Paarungen ermittelt. In einigen Ausführungsformen wird ein Bit generiert, das den Sieger jedes Wettlaufs in jeder Runde repräsentiert. In einigen Ausführungsformen generieren die Paarungen der ersten Runde und der zweiten Runde eine Anzahl von vollen Entropiebits, die gleich der Anzahl der PUF-Zellen ist.

[0104] Bei Handlung **909** wird eine PUF-Ausgabe auf Grundlage der Ergebnisse der ersten Runde und der Ergebnisse der zweiten Runde generiert. Die PUF-Ausgabe kann eine beliebige geeignete Anzahl von Bits (z. B. 32, 64, 128 oder 256 Bits) aufweisen. Die PUF-Ausgabe kann verwendet werden, um ein Geheimnis zu verschleiern - (um z. B. Abschnitte des Geheimnisses so zu verdecken, zu codieren und/oder zu verschlüsseln, dass das Ergebnis der Verschleierung öffentlich gemacht werden kann, ohne das Geheimnis zu enthüllen). Die PUF-Ausgabe kann auch verwendet werden, um ein Geheimnis auf Grundlage von Hilfsdaten erneut zu generieren, die während der Verschleierung des Geheimnisses erstellt wurden.

[0105] In weiteren Ausführungsformen kann der Prozess für drei oder mehr Runden fortfahren. In einigen Ausführungsformen ordnen ein oder mehrere Prozessoren oder Steuerverschaltungen in einer auf die zweite Runde folgenden dritten Runde dritte Paarungen von jeweiligen der Vielzahl der PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde und der zweiten Runde zu, wobei jede PUF-Zelle der Vielzahl der PUF-Zellen mit einer PUF-Zelle der Vielzahl der PUF-Zellen gepaart wird, die der Sieger einer gleichen Anzahl von Runden (oder Paarungen) war. In der dritten Runde können jeweilige dritte Ausgaben von jeder der Vielzahl von PUF-Zellen in der dritten Runde generiert werden, um einen Sieger für jedes Paar von PUF-Zellen in den dritten Paarungen zu ermitteln. In einigen Ausführungsformen können zusätzliche Runden von Paarungen (z. B. eine vierte Runde für eine 64-PUF-Zellen-Implementierung oder eine fünfte Runde) auf Grundlage der Ergebnisse von vorangehenden Runden zugeordnet werden. In einigen Ausführungsformen ist die Gesamtanzahl an Runden durch mindestens einen Prozessor eingeschränkt, sodass sie den binären Logarithmus der Anzahl von PUF-Zellen in der Vielzahl der PUF-Zellen nicht überschreitet. In weiteren Ausführungsformen kann die PUF-Ausgabe eine Anzahl von vollen Bits von Entropie aufweisen, die die Anzahl der PUF-Zellen überschreitet.

[0106] In einigen Beispielen können die hierin offenbarten Komponenten Parameter oder Anweisungen lesen, die die von den Komponenten durchgeführten Funktionen beeinflussen. Diese Parameter oder Anweisungen können physisch in einer beliebigen Form von geeignetem Speicher einschließlich flüchtigem Speicher (wie RAM) oder nichtflüchtigem Speicher (wie eine magnetische Festplatte) gespeichert sein. Darüber hinaus können die Parameter oder Anweisungen logisch in einer proprietären Datenstruktur (wie einer Datenbank oder einer Datei, die von einer Anwendung im Benutzerraum definiert wird) oder in einer gemeinsam genutzten Datenstruktur (wie einer Anwendungsregistrierungsdatenbank, die durch ein Betriebssystem definiert wird) gespeichert sein. Darüber hinaus bieten einige Beispiele sowohl System- als auch Benutzeroberflächen, die es externen Entitäten ermöglichen, die Parameter und Anweisungen zu ändern und dadurch das Verhalten der Komponenten zu konfigurieren.

[0107] Basierend auf der vorstehenden Offenbarung sollte es für Durchschnittsfachleute offensichtlich sein, dass die hierin offenbarten Ausführungsformen nicht auf eine bestimmte Computersystemplattform, einen Prozessor, ein Betriebssystem, ein Netzwerk oder ein Kommunikationsprotokoll beschränkt sind. Es sollte auch offensichtlich sein, dass die hierin offenbarten Ausführungsformen nicht auf eine bestimmte Architektur beschränkt sind.

[0108] Es versteht sich, dass Ausführungsformen der hier diskutierten Verfahren und Vorrichtungen nicht auf die Details der Konstruktion und die Anord-

nung der Komponenten beschränkt sind, die in der folgenden Beschreibung dargelegt oder in den beigefügten Zeichnungen dargestellt sind. Die Verfahren und Einrichtungen sind in anderen Ausführungsformen implementierbar und können auf verschiedene Weise praktiziert oder ausgeführt werden. Beispiele für spezifische Implementierungen werden hier nur zur Veranschaulichung bereitgestellt und sollen nicht einschränkend sein. Insbesondere sollen Handlungen, Elemente und Merkmale, die in Verbindung mit einer oder mehreren Ausführungsformen diskutiert werden, nicht von einer ähnlichen Rolle in anderen Ausführungsformen ausgeschlossen werden.

[0109] Die hierin verwendete Ausdrucksweise und Terminologie dient ebenfalls der Beschreibung und sollte nicht als einschränkend betrachtet werden. Alle Verweise auf Ausführungsformen oder Elemente oder Handlungen der Systeme und Verfahren, auf die hierin im Singular Bezug genommen wird, können auch Ausführungsformen aufweisen, die eine Vielzahl dieser Elemente aufweisen, und alle Verweise auf mehrere Ausführungsformen oder Elemente oder Handlungen hierin können auch Ausführungsformen einschließen, die nur ein einzelnes Element aufweisen. Bezugnahmen in der Singular- oder Pluralform sollen die vorliegend offenbarten Systeme oder Verfahren, ihre Komponenten, Handlungen oder Elemente nicht einschränken. Die Verwendung von „enthalten“, „umfassen“, „aufweisen“, „beinhalten“, „involvieren“ und Variationen davon soll hierin die danach aufgelisteten Elemente und deren Äquivalente sowie zusätzliche Elemente aufweisen. Verweise auf „oder“ können als inklusiv ausgelegt werden, sodass alle mit „oder“ beschriebenen Begriffe einen, mehrere oder alle beschriebenen Begriffe anzeigen können. Die Verwendung von mindestens einem der Elemente und einer Liste von Elementen (z. B. A, B, C) soll eine Auswahl aus A, B, C (z. B. A), zwei beliebige Auswahlen aus A, B, C (z. B. A und B), drei beliebige Auswahlen (z. B. A, B, C) usw. und beliebige Vielfache jeder Auswahl abdecken.

[0110] Nachdem somit mehrere Aspekte von mindestens einer Ausführungsform dieser Erfindung beschrieben worden sind, sind verschiedene Änderungen, Modifikationen und Verbesserungen für Fachleute ohne Weiteres ersichtlich. Derartige Änderungen, Modifikationen und Verbesserungen sollen Teil dieser Offenbarung sein und sollen innerhalb des Umfangs der Erfindung liegen. Dementsprechend sind die vorstehende Beschreibung und die Zeichnungen nur beispielhaft.

[0111] Nach verschiedenen Gesichtspunkten ist eine Vorrichtung mit einer verzögerungsbasierten physikalisch unklonbaren Funktion (PUF) vorgesehen. Nach einer Ausführungsform weist die PUF-Vorrichtung Verschaltung zum Generieren von Entropie-Ausgabebits durch Vergleichen oder „in einem Wettlauf Durchlaufen“ einer Vielzahl von PUF-Zellen auf. Eine PUF-Zelle ist ein Baustein der PUF-Vorrichtung. Die PUF-Vorrichtung kann zum Beispiel zwei identisch konstruierte Schaltkreise mit nur prozessbedingten Schwankungen aufweisen und jeder Schaltkreis kann eine PUF-Zelle sein. Nach einem anderen Gesichtspunkt, falls PUF-Zellen mit einem gleichen Sieg- oder Niederlagen-Verlauf in einem Wettlauf verglichen werden, können Angreifer das Ergebnis des aktuellen Wettlaufs nicht aufgrund von vorangehenden Wettlaufergebnissen vorhersagen. Dementsprechend werden hierin Systeme und Verfahren zum Generieren von mehreren Wettlaufrunden auf Grundlage der vorangehenden Wettlaufrunden beschrieben. Deshalb kann eine PUF-Zelle in mehreren paarweisen Vergleichen verwendet werden, während eine maximale Entropie extrahiert wird.

Patentansprüche

1. Verfahren zum Generieren von Entropie in einer Codierung mit einer physikalisch unklonbaren Funktion (PUF), wobei das Verfahren aufweist:
Zuordnen von ersten Paarungen von jeweiligen aus einer Vielzahl von PUF-Zellen in einer ersten Runde durch mindestens einen Prozessor;
Generieren von jeweiligen ersten Ausgaben von jeder der Vielzahl von PUF-Zellen in der ersten Runde und Ermitteln von Ergebnissen für die erste Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den ersten Paarungen;
Zuordnen von zweiten Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer zweiten, auf die erste Runde folgenden Runde;
Generieren von jeweiligen zweiten Ausgaben von jeder der Vielzahl von PUF-Zellen in der zweiten Runde und Ermitteln von Ergebnissen für die zweite Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den zweiten Paarungen; und
Generieren einer PUF-Ausgabe auf Grundlage der Ergebnisse der ersten Runde und der Ergebnisse der zweiten Runde.

2. Verfahren nach Anspruch 1, das ferner ein Verschleiern eines Geheimnisses unter Verwendung der PUF-Ausgabe aufweist.

3. Verfahren nach einem der vorangehenden Ansprüche, wobei jede PUF-Zelle der Vielzahl der PUF-Zellen einen von einer Vielzahl von identisch konstruierten Schaltkreisen mit Unterschieden aufweist, die von Schwankungen im Fertigungsprozess stammen.

4. Verfahren nach einem der vorangehenden Ansprüche, wobei das Zuordnen von zweiten Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer zweiten, auf die erste Runde folgenden Runde aufweist:

Zuordnen von Paarungen der jeweiligen Sieger aus jedem Paar von PUF-Zellen in der ersten Paarung; und
Zuordnen von Paarungen der restlichen PUF-Zellen, die in der ersten Paarung keine Sieger waren.

5. Verfahren nach einem der vorangehenden Ansprüche, ferner aufweisend:
Zuordnen, in einer auf die zweite Runde folgenden dritten Runde, von dritten Paarungen von jeweiligen der Vielzahl der PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde und der zweiten Runde, wobei jede PUF-Zelle der Vielzahl der PUF-Zellen mit einer PUF-Zelle der Vielzahl der PUF-Zellen gepaart wird, die der Sieger einer gleichen Anzahl von Runden war; und
Generieren von jeweiligen dritten Ausgaben von jeder der Vielzahl von PUF-Zellen in der dritten Runde und Ermitteln eines Siegers für jedes Paar von PUF-Zellen in den dritten Paarungen.

6. Verfahren nach einem der vorangehenden Ansprüche, wobei ein erster Sieger eines ersten Paars in einer vorangehenden Runde mit einem zweiten Sieger eines zweiten Paars in der vorangehenden Runde gepaart wird.

7. Verfahren nach einem der vorangehenden Ansprüche, das ferner ein Zuordnen zusätzlicher Runden von PUF-Zellenpaarungen auf Grundlage der Ergebnisse von vorangehenden Runden aufweist, durch:
Gruppieren von PUF-Zellen, die in einer vorangehenden Runde gepaart wurden, in eine Vielzahl von Gruppen; und
Paaren jeder PUF-Zelle in einer ersten Gruppe mit einer jeweiligen PUF-Zelle in einer zweiten Gruppe.

8. Verfahren nach einem der vorangehenden Ansprüche, ferner aufweisend ein Generieren eines Bits, das den Sieger jedes Paars von PUF-Zellen in jeder Runde repräsentiert, wobei ein Sieger eines Paars von PUF-Zellen mit einer Ausgabe eines binären Vergleichs der jeweiligen Ausgaben jeder PUF-Zelle im Paar assoziiert ist und wobei femer die Paarungen der ersten Runde und der zweiten Runde eine Anzahl von Entropiebits generieren, die gleich der Anzahl der PUF-Zellen ist.

9. Verfahren nach einem der vorangehenden Ansprüche, ferner aufweisend ein Zuordnen zusätzlicher Runden von Paarungen auf Grundlage der Ergebnisse von vorangehenden Runden, wobei die Gesamtanzahl der Runden durch den mindestens einen Prozessor so eingeschränkt ist, dass sie den binären Logarithmus der Anzahl der PUF-Zellen in der Vielzahl der PUF-Zellen nicht überschreitet.

10. Verfahren nach einem der vorangehenden Ansprüche, wobei das Verfahren ferner aufweist:
Anwenden eines Fehlerkorrekturcodes auf die jeweiligen ersten Ausgaben von jeder der Vielzahl der PUF-Zellen vor der zweiten Runde durch den mindestens einen Prozessor.

11. System zum Generieren von Entropie in einer Codierung mit einer physikalisch unklonbaren Funktion (PUF), wobei das System aufweist:
eine Vielzahl von PUF-Zellen; und
mindestens einen Prozessor, der ausgebildet ist:
erste Paarungen von jeweiligen aus einer Vielzahl von PUF-Zellen in einer ersten Runde zuzuordnen;
jeweilige erste Ausgaben von jeder der Vielzahl von PUF-Zellen in der ersten Runde zu generieren und Ergebnisse für die erste Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den ersten Paarungen zu ermitteln;
zweite Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer zweiten, auf die erste Runde folgenden Runde zuzuordnen;
jeweilige zweite Ausgaben von jeder der Vielzahl von PUF-Zellen in der zweiten Runde zu generieren und Ergebnisse für die zweite Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den zweiten Paarungen zu ermitteln; und
eine PUF-Ausgabe auf Grundlage der Ergebnisse der ersten Runde und der Ergebnisse der zweiten Runde zu generieren.

12. System nach Anspruch 11, wobei der mindestens eine Prozessor ferner ausgebildet ist, ein Geheimnis unter Verwendung der PUF-Ausgabe zu verschleiern.

13. System nach einem der Ansprüche 11 und 12, wobei jede PUF-Zelle der Vielzahl der PUF-Zellen einen von einer Vielzahl von identisch konstruierten Schaltkreisen mit Unterschieden aufweist, die von Schwankungen im Fertigungsprozess stammen.

14. System nach einem der Ansprüche 11 bis 13, wobei das Zuordnen von zweiten Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer zweiten, auf die erste Runde folgenden Runde aufweist:
Zuordnen von Paarungen der jeweiligen Sieger aus jedem Paar von PUF-Zellen in der ersten Paarung; und
Zuordnen von Paarungen der restlichen PUF-Zellen, die in der ersten Paarung keine Sieger waren.

15. System nach einem der Ansprüche 11 bis 14, wobei der mindestens eine Prozessor ferner ausgebildet ist:
in einer auf die zweite Runde folgenden dritten Runde dritte Paarungen von jeweiligen der Vielzahl der PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde und der zweiten Runde zuzuordnen, wobei jede PUF-Zelle der Vielzahl der PUF-Zellen mit einer

PUF-Zelle der Vielzahl der PUF-Zellen gepaart wird, die der Sieger einer gleichen Anzahl von Runden war; und
jeweilige dritte Ausgaben von jeder der Vielzahl von PUF-Zellen in der dritten Runde zu generieren und einen Sieger für jedes Paar von PUF-Zellen in den dritten Paarungen zu ermitteln.

16. System nach einem der Ansprüche 11 bis 15, wobei ein erster Sieger eines ersten Paars in einer vorangehenden Runde mit einem zweiten Sieger eines zweiten Paars in der vorangehenden Runde gepaart wird und wobei ferner der mindestens eine Prozessor ferner ausgebildet ist, vor einer aktuellen Runde einen Fehlerkorrekturcode auf jeweilige Ausgaben von jeder der Vielzahl der PUF-Zellen anzuwenden.

17. System nach einem der Ansprüche 11 bis 16, wobei der mindestens eine Prozessor ferner ausgebildet ist, zusätzliche Runden von PUF-Zellenpaarungen auf Grundlage der Ergebnisse von vorangehenden Runden zuzuordnen, durch:
Gruppieren von PUF-Zellen, die in einer vorangehenden Runde gepaart wurden, in eine Vielzahl von Gruppen; und
Paaren jeder PUF-Zelle in einer ersten Gruppe mit einer jeweiligen PUF-Zelle in einer zweiten Gruppe.

18. System nach einem der Ansprüche 11 bis 17, ferner aufweisend einen binären Komparator, wobei der mindestens eine Prozessor ferner ausgebildet ist, ein Bit zu generieren, das den Sieger jedes Paars von PUF-Zellen in jeder Runde repräsentiert, wobei ein Sieger eines Paars von PUF-Zellen mit einer Ausgabe des binären Komparators assoziiert ist, der jeweilige Ausgaben jeder PUF-Zelle im Paar verglichen hat, und wobei ferner die Paarungen der ersten Runde und der zweiten Runde eine Anzahl von Entropiebits generieren, die gleich der Anzahl der PUF-Zellen ist.

19. System nach einem der Ansprüche 11 bis 18, wobei der mindestens eine Prozessor ferner ausgebildet ist, zusätzliche Runden von Paarungen auf Grundlage der Ergebnisse von vorangehenden Runden zuzuordnen, wobei die Gesamtanzahl der Runden durch den mindestens einen Prozessor so eingeschränkt ist, dass sie den binären Logarithmus der Anzahl der PUF-Zellen in der Vielzahl der PUF-Zellen nicht überschreitet.

20. Nichtflüchtiges computerlesbares Medium oder mehrere nichtflüchtige computerlesbare Medien, das/die prozessorausführbare Anweisungen speichert/speichern, die, wenn sie ausgeführt werden, bewirken, dass mindestens ein Prozessor ein Verfahren durchführt, aufweisend:
Zuordnen von ersten Paarungen von jeweiligen aus einer Vielzahl von PUF-Zellen in einer ersten Runde;
Generieren von jeweiligen ersten Ausgaben von jeder der Vielzahl von PUF-Zellen in der ersten Runde und Ermitteln von Ergebnissen für die erste Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den ersten Paarungen;
Zuordnen von zweiten Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer zweiten, auf die erste Runde folgenden Runde;
Generieren von jeweiligen zweiten Ausgaben von jeder der Vielzahl von PUF-Zellen in der zweiten Runde und Ermitteln von Ergebnissen für die zweite Runde einschließlich eines Siegers für jedes Paar von PUF-Zellen in den zweiten Paarungen; und
Generieren einer PUF-Ausgabe auf Grundlage der Ergebnisse der ersten Runde und der Ergebnisse der zweiten Runde.

Es folgen 9 Seiten Zeichnungen

Anhängende Zeichnungen

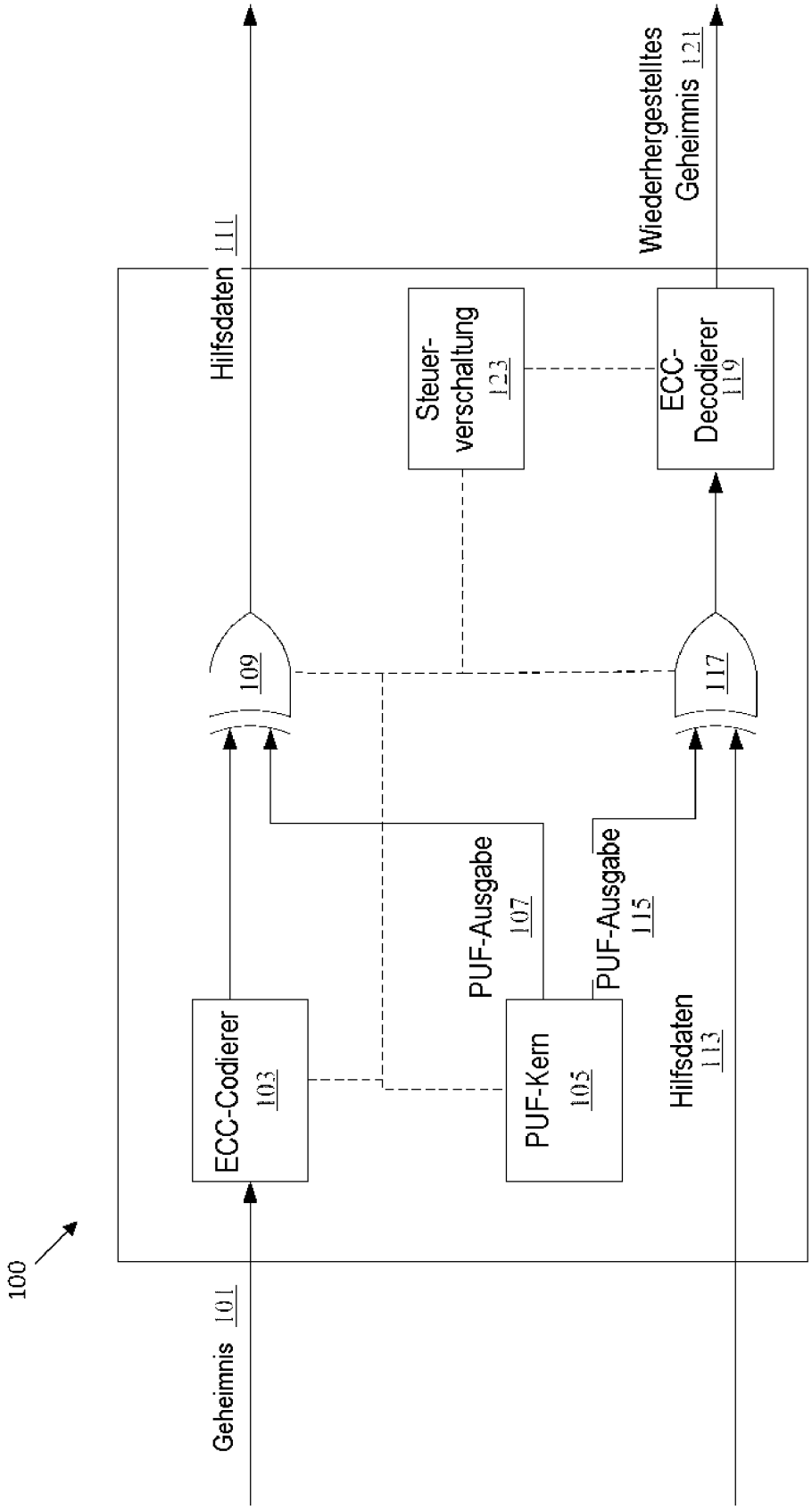


FIG. 1

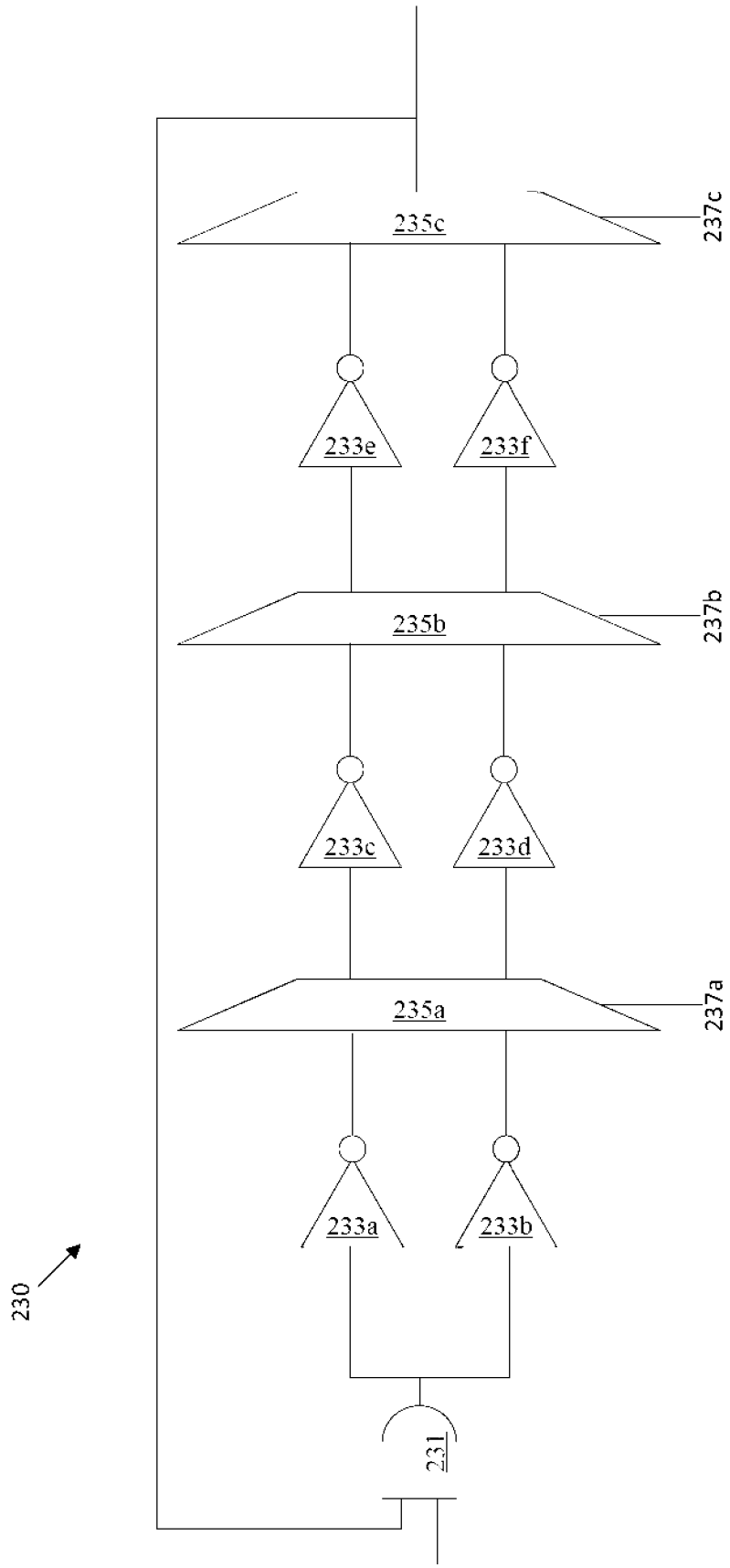

230
231
233a
233b
233c
233d
233e
233f
235a
235b
235c
237a
237b
237c

FIG. 2A

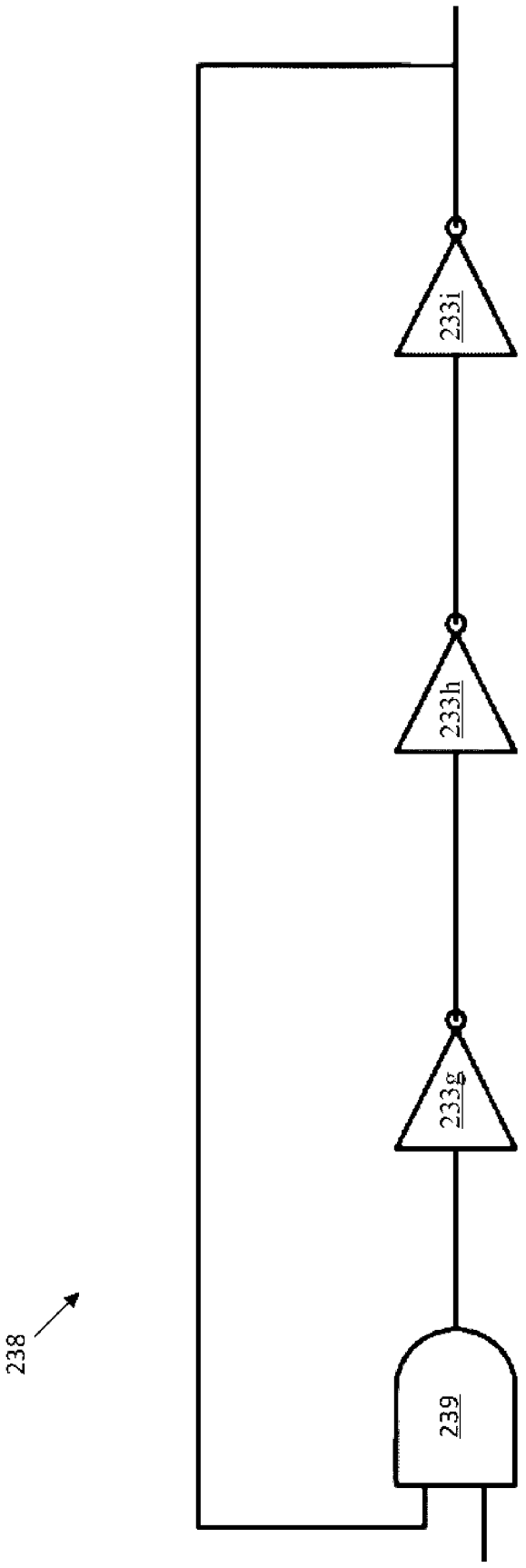


FIG. 2B

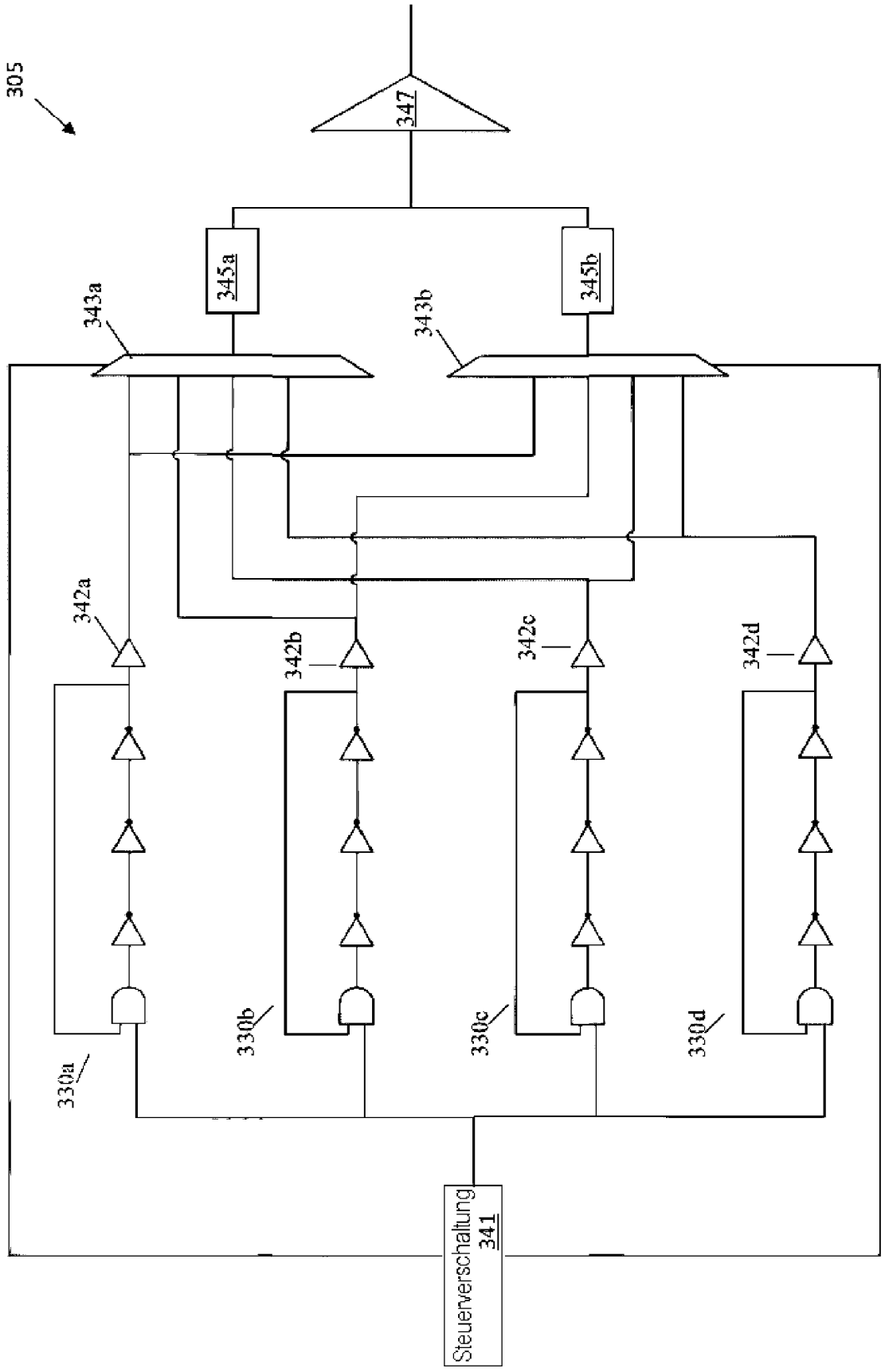

305
347
345a
345b
343a
343b
342a
342b
342c
342d
330a
330b
330c
330d
Steuerverschaltung
341

FIG. 3

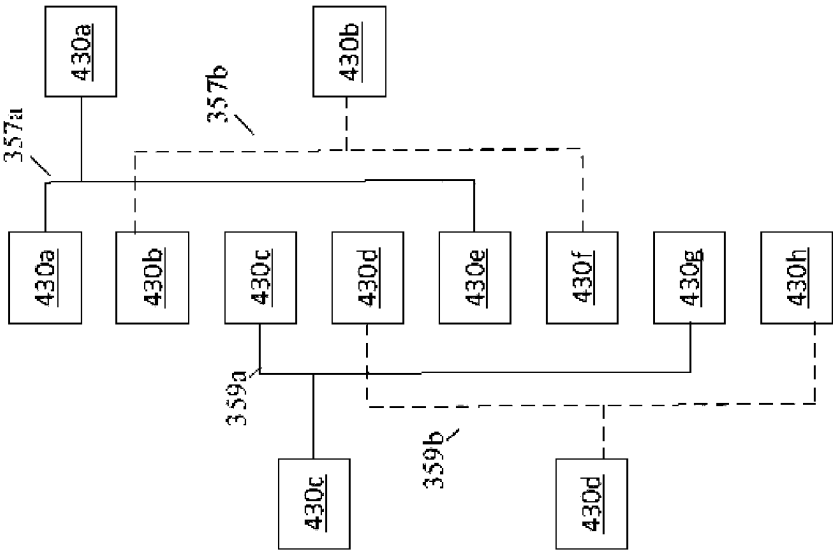


FIG. 4C

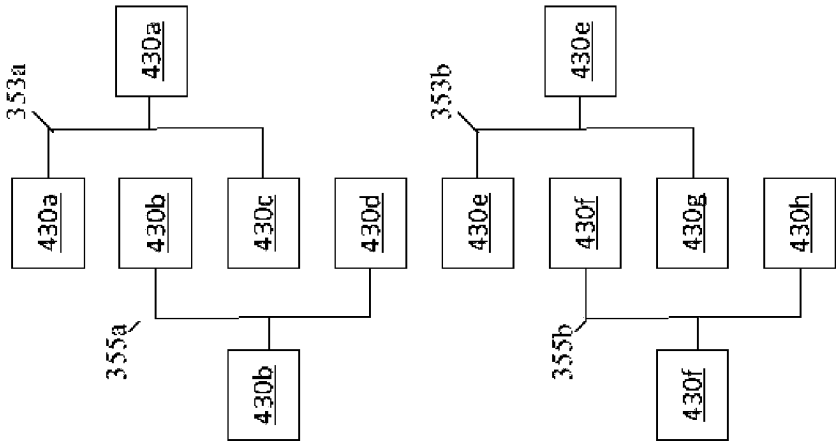


FIG. 4B

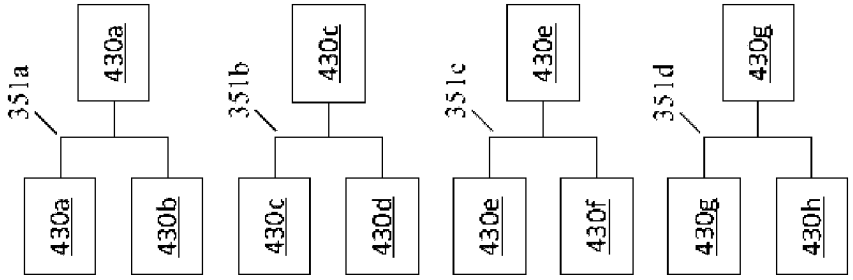


FIG. 4A

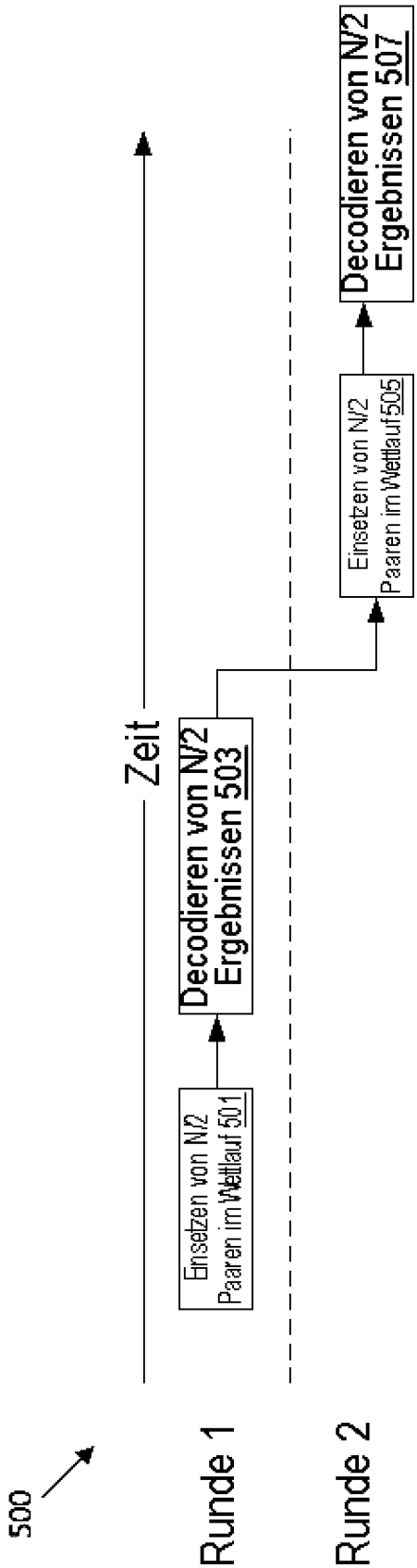

500
Zeit
Runde 1
Runde 2
Einsetzen von N/2 Paaren im Wettlauf 501
Decodieren von N/2 Ergebnissen 503
Einsetzen von N/2 Paaren im Wettlauf 505
Decodieren von N/2 Ergebnissen 507

FIG. 5A

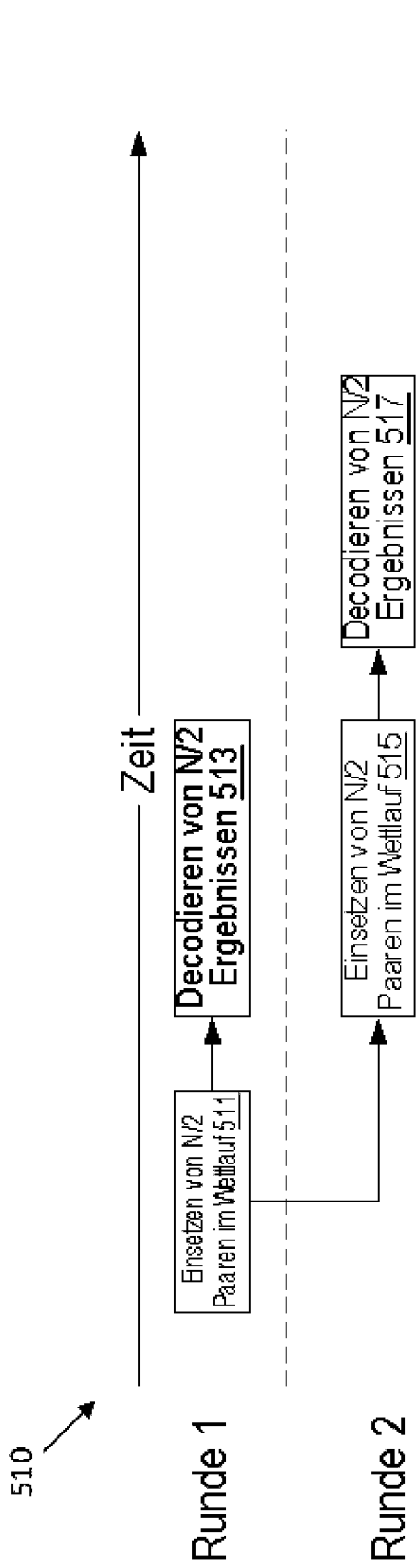

510
Zeit
Runde 1
Runde 2
Einsetzen von N/2 Paaren im Wettlauf 511
Decodieren von N/2 Ergebnissen 513
Einsetzen von N/2 Paaren im Wettlauf 515
Decodieren von N/2 Ergebnissen 517

FIG. 5B

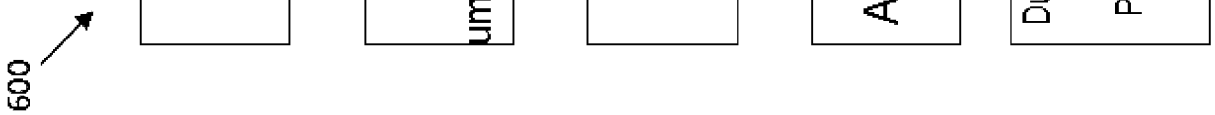

600
Erhalten von Geheimnis K 601
Anwenden von Fehlerkorrekturcode auf K, um ECC-Daten zu erzeugen 603
Anfügen von ECC-Daten an K 605
Generieren einer PUF-Ausgabe unter Verwendung eines PUF-Kerns 607
Durchführen einer Operation, um K und ECC-Daten unter Verwendung der PUF-Ausgabe zu verschleiern, um ein Ausgabewort W zu erstellen 609

FIG. 6

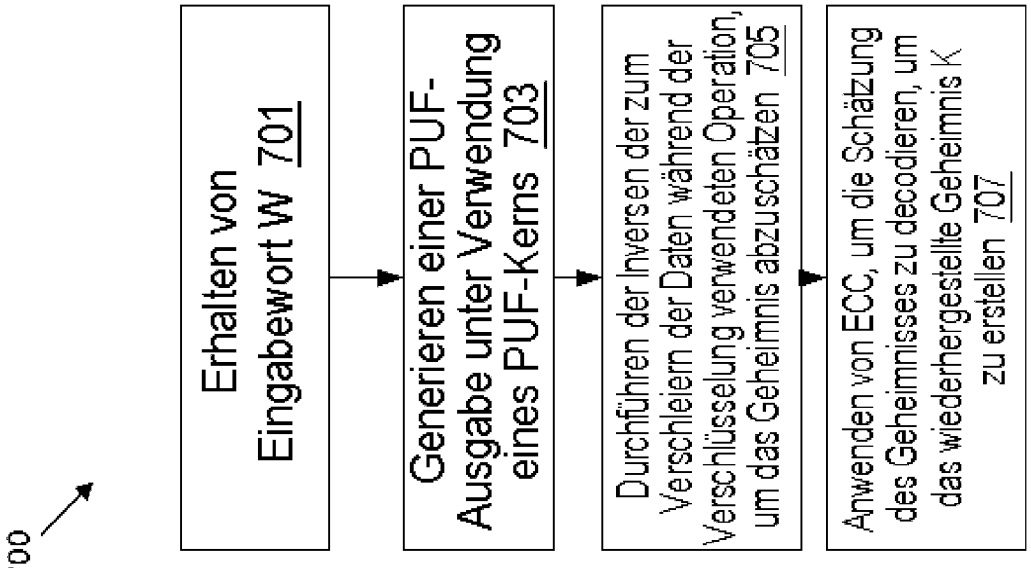

700
Erhalten von Eingabewort W 701
Generieren einer PUF-Ausgabe unter Verwendung eines PUF-Kerns 703
Durchführen der Inversen der zum Verschleiern der Daten während der Verschlüsselung verwendeten Operation, um das Geheimnis abzuschätzen 705
Anwenden von ECC, um die Schätzung des Geheimnisses zu decodieren, um das wiederhergestellte Geheimnis K zu erstellen 707

FIG. 7

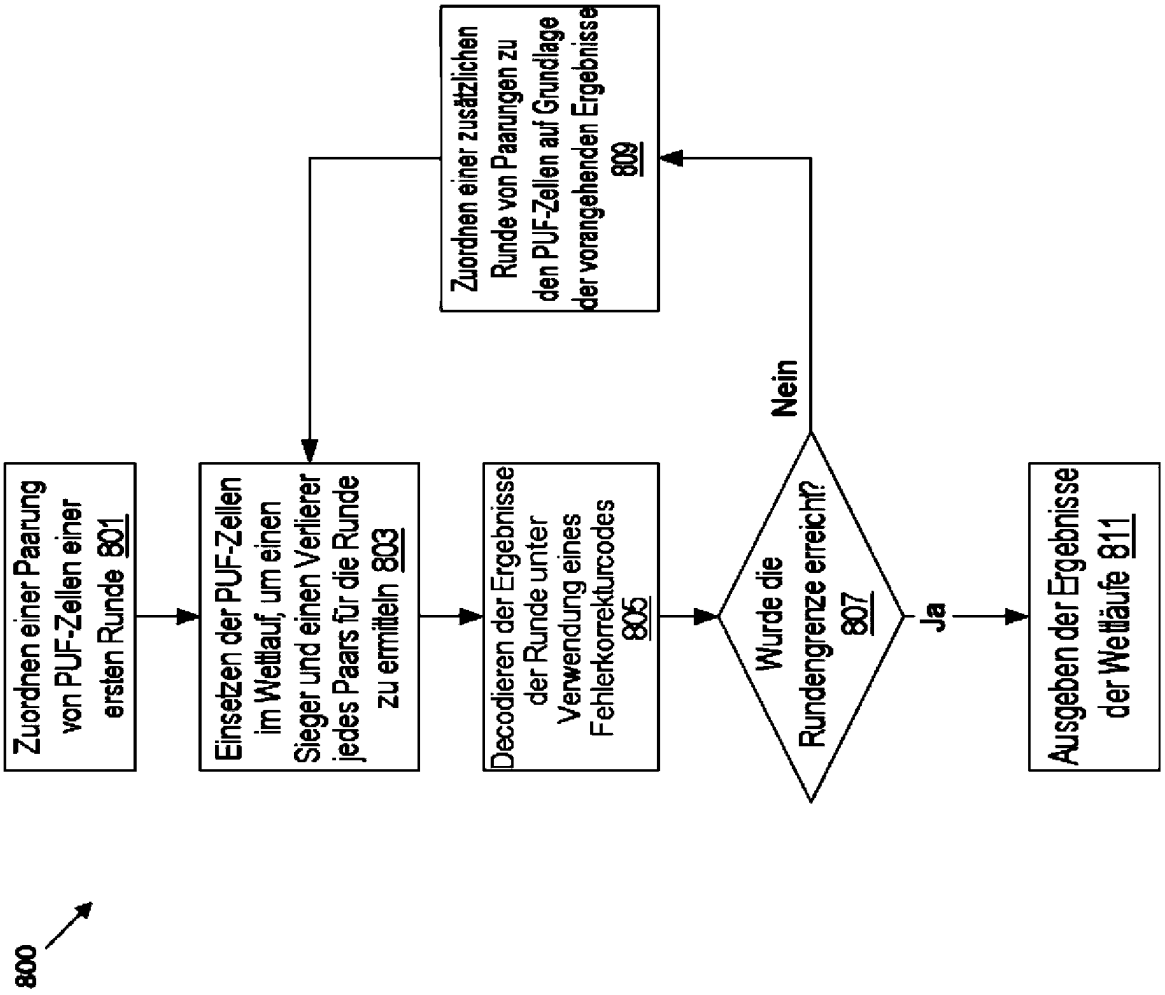

800
Zuordnen einer Paarung von PUF-Zellen einer ersten Runde 801
Einsetzen der PUF-Zellen im Wettlauf, um einen Sieger und einen Verlierer jedes Paars für die Runde zu ermitteln 803
Decodieren der Ergebnisse der Runde unter Verwendung eines Fehlerkorrekturcodes 805
Wurde die Rundengrenze erreicht? 807
Nein
Ja
Zuordnen einer zusätzlichen Runde von Paarungen zu den PUF-Zellen auf Grundlage der vorangehenden Ergebnisse 809
Ausgeben der Ergebnisse der Wettläufe 811

FIG. 8

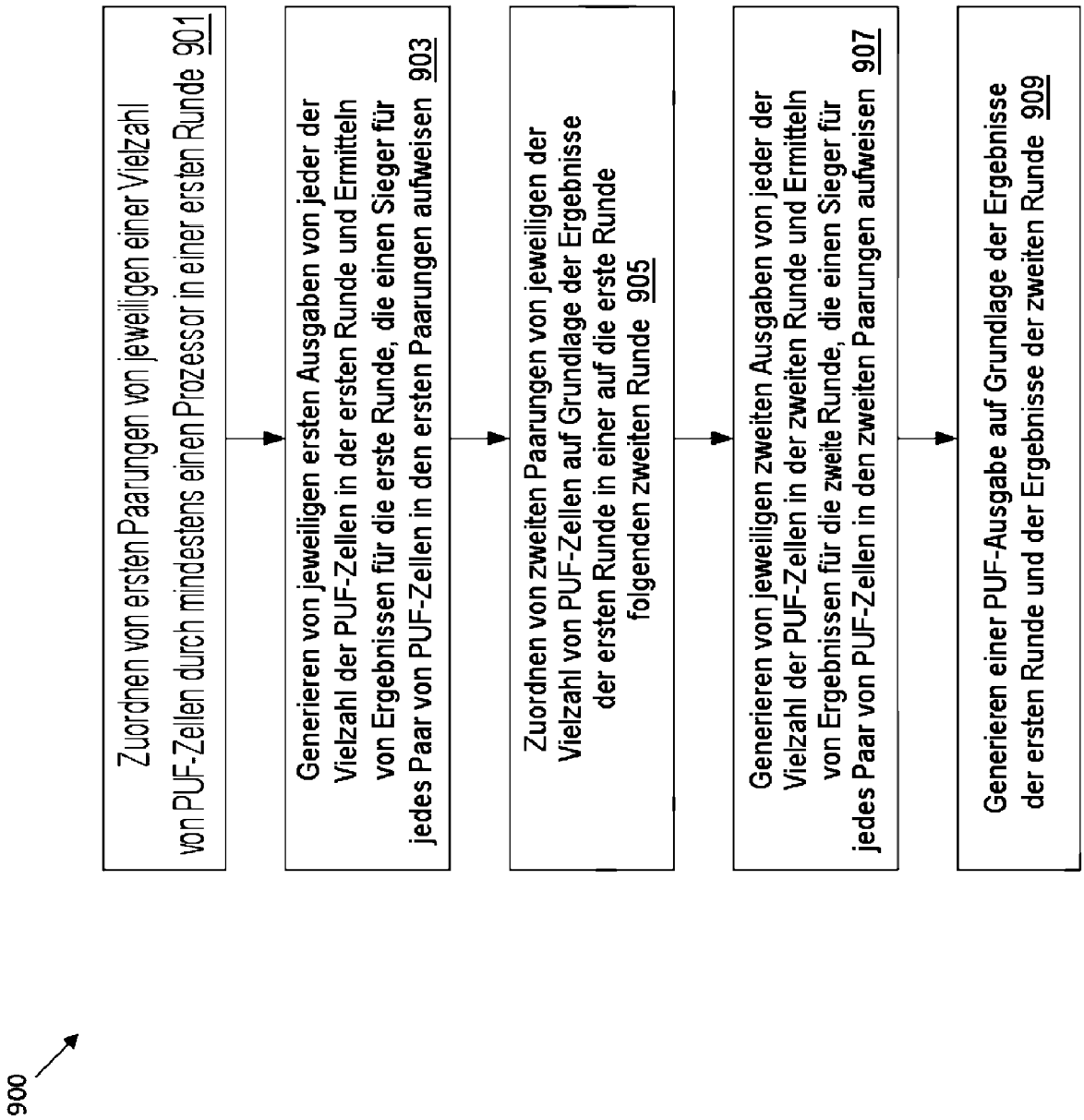

900
Zuordnen von ersten Paarungen von jeweiligen einer Vielzahl von PUF-Zellen durch mindestens einen Prozessor in einer ersten Runde 901
Generieren von jeweiligen ersten Ausgaben von jeder der Vielzahl der PUF-Zellen in der ersten Runde und Ermitteln von Ergebnissen für die erste Runde, die einen Sieger für jedes Paar von PUF-Zellen in den ersten Paarungen aufweisen 903
Zuordnen von zweiten Paarungen von jeweiligen der Vielzahl von PUF-Zellen auf Grundlage der Ergebnisse der ersten Runde in einer auf die erste Runde folgenden zweiten Runde 905
Generieren von jeweiligen zweiten Ausgaben von jeder der Vielzahl der PUF-Zellen in der zweiten Runde und Ermitteln von Ergebnissen für die zweite Runde, die einen Sieger für jedes Paar von PUF-Zellen in den zweiten Paarungen aufweisen 907
Generieren einer PUF-Ausgabe auf Grundlage der Ergebnisse der ersten Runde und der Ergebnisse der zweiten Runde 909

FIG. 9