

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2012年9月13日(13.09.2012)



(10) 国際公開番号
WO 2012/120671 A1

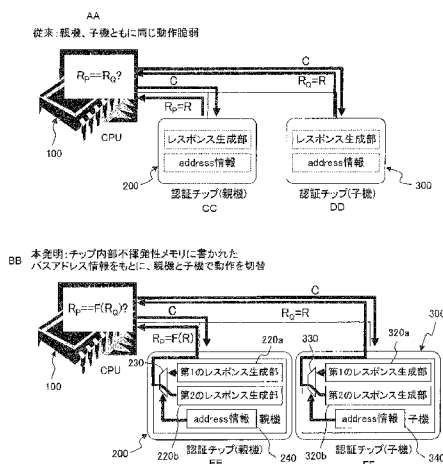
- (51) 国際特許分類:
H04L 9/32 (2006.01)
- (21) 国際出願番号: PCT/JP2011/055557
- (22) 国際出願日: 2011年3月9日(09.03.2011)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (71) 出願人(米国を除く全ての指定国について): 富士通株式会社(FUJITSU LIMITED) [JP/JP]; 〒2118588 神奈川県川崎市中原区上小田中4丁目1番1号 Kanagawa (JP).
- (72) 発明者; および
- (75) 発明者/出願人(米国についてのみ): 伊藤孝一(ITO, Kouichi) [JP/JP]; 〒2118588 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内 Kanagawa (JP). 武仲正彦(TAKENAKA, Masahiko) [JP/JP]; 〒2118588 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内 Kanagawa (JP).
- (74) 代理人: 大菅義之(OSUGA, Yoshiyuki); 〒1020084 東京都千代田区二番町8番地20二番町ビル3F Tokyo (JP).
- (81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT,

[続葉有]

(54) Title: AUTHENTICATION SYSTEM USING SYMMETRIC-KEY CRYPTOGRAPHY

(54) 発明の名称: 共通鍵暗号を用いた認証システム

[図9]



- 200 Response generating unit
Address information
300 Response generating unit
Address information
220a First response generating unit
220b Second response generating unit
320a First response generating unit
320b Second response generating unit
240 Address information
Master device
340 Address information
Slave device
AA Prior art: Master and slave devices have same operation: vulnerable
BB Present invention: Operation switched between master and slave devices on the basis of bus address information written in nonvolatile memory inside chip
CC Authentication chip (master device)
DD Authentication chip (slave device)
EE Authentication chip (master device)
FF Authentication chip (slave device)

(57) Abstract: The present invention provides a method of authentication wherein a processor controlling a master device authenticates a slave device. The method of authenticating the slave device comprises: generating a random number; transmitting the random number to a master device authentication chip and a slave device authentication chip; receiving from the master device authentication chip a first response value which is obtained by applying a first conversion function determined by a value set in the master device authentication chip to an output value generated by applying a cryptographic function to an integer stored in the master device authentication chip and the random number; receiving from the slave device authentication chip a second response value which is obtained by applying a second conversion function determined by a value set in the slave device authentication chip to an output value generated by applying a cryptographic function to an integer stored in the slave device authentication chip and the random number; and comparing the second response value with a value obtained by applying to the first response value a third conversion function determined by the number of difference between the value set in the master device authentication chip and the value set in the slave device authentication chip.

(57) 要約:

[続葉有]

WO 2012/120671 A1



NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI 添付公開書類:

(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, — 国際調査報告 (条約第 21 条(3))
NE, SN, TD, TG).

親機を制御するプロセッサが子機を認証する方法であって、乱数を生成し；前記乱数を前記親機認証チップおよび前記子機認証チップに送信し；前記親機認証チップに記憶された整数および前記乱数に対して暗号化関数を作用させて生成された出力値に、前記親機認証チップに設定された値で決まる第 1 の変換関数を作用させて得られる第 1 のレスポンス値を前記親機認証チップから受信し；前記子機認証チップに記憶された前記整数および前記乱数に対して暗号化関数を作用させて生成された前記出力値に、前記子機認証チップに設定された値で決まる第 2 の変換関数を作用させて得られる第 2 のレスポンス値を前記子機認証チップから受信し；前記第 1 のレスポンス値に、前記親機認証チップに設定された値と前記子機認証チップに設定された値の大きさの差の数で決まる第 3 の変換関数を作用させて得られる値と前記第 2 のレスポンス値を比較して前記子機を認証する方法。

明 細 書

発明の名称： 共通鍵暗号を用いた認証システム

技術分野

[0001] 本発明は、共通鍵暗号による電子デバイスの認証装置および方法に関し、特に共通鍵暗号機能を備えた組み込み機器の偽造防止を行うための認証システム技術に関するものである。

背景技術

[0002] プリンタカートリッジや医療機器の分野で、正規品と偽造品を区別するために、認証機能を持つ組み込み機器を用いることで、粗悪な偽造品により発生する被害から顧客を守ることができる。すなわち、認証機能を持つ組み込み機器の偽造品を防ぐことが、製品を利用する顧客の利益を守ることに直結するため、認証機能を持つ組み込み機器の偽造防止は、産業上の非常に重要な課題である。

[0003] 認証においては、暗号方式がコア機能として使用される。暗号方式は、公開鍵暗号方式と共通鍵暗号方式に大別される。公開鍵暗号方式とは、暗号化と復号で異なる鍵を用いる方式であり、暗号化を行うための鍵（公開鍵）を一般に公開する代わりに、暗号文を復号するための鍵（秘密鍵）を受信者のみの秘密情報とすることで安全性を保つ方式である。これに対し、共通鍵暗号方式と呼ばれるものは、暗号化と復号で同一の鍵（秘密鍵）を用いる方式であり、この秘密鍵を送信者と受信者以外の第三者にわからない情報とすることで安全性を保つ方式である。いずれの暗号方式を用いた場合でも、秘密鍵が外部に漏洩しないことがセキュリティ上の大きな前提となる。この前提が守られる限りは、公開鍵暗号、共通鍵暗号のいずれを用いた認証も安全となるが、共通鍵暗号の方が公開鍵暗号より回路規模が小さく、低コスト化には適しているため、特に組み込み機器においては共通鍵暗号を用いた認証が広く普及している。

[0004] セキュリティ機能を備えた組み込み機器に求められる性質として、耐タン

パ性と呼ばれるものがある。耐タンパとは、覗き見防止の意味であり、組み込み機器の内部に格納されている秘密鍵等の重要情報を、外部から不正に漏洩させるのが困難な性質をさす。耐タンパ性を実現するためには、組み込み機器が備える正規の入出力端子からのアクセスによる情報漏洩防止はもちろんのこと、マイクロプローブを用いて内部回路を直接覗き込む非正規アクセスに対する情報漏洩防止も実現しなければならない。通常のＩＣチップのハードウェア構成では、マイクロプローブを用いて内部回路を直接覗き込む攻撃に耐えることはできない。よって、耐タンパ性を実現するには、外部からの物理的・論理的な不正アクセスを防ぐための専用のハードウェア構成をもったＩＣチップとして製造する必要がある。

- [0005] 安全な認証を実現するための一般的な方法としては、耐タンパ性をもつ認証機器に対して、機器製造時に認証に用いる秘密鍵を内部に書き込み、製造後は鍵を外部に取り出さない方法が用いられる。こうすることで、不正な第三者が秘密鍵を入手し、認証機器の偽造を防止することができる。耐タンパ性をもつ認証機器に対して、秘密鍵を書き込み、鍵を外部に取り出すことがない状態で、認証機器間で認証プロトコルを実行することで、安全な認証を実現し、粗悪な偽造品により発生する被害から顧客を守ることができると考えられている。

発明の概要

発明が解決しようとする課題

- [0006] 共通鍵暗号を用いた認証において、秘密鍵が外部に漏洩しない、生産コストを削減する、通信トポロジをなるべく簡素にするという３つの制約下での安全な認証を実現するのは困難であり、これらの制約下でも偽造品をふせぐ安全でコンパクトな認証プロトコルおよびシステムが望まれている。

課題を解決するための手段

- [0007] 親機認証チップを含む親機を制御するプロセッサが子機認証チップを含む子機を認証する方法は：乱数を生成することと；前記乱数を前記親機認証チップおよび前記子機認証チップに送信することと；前記親機認証チップに秘

密鍵として記憶された整数および前記乱数に対して暗号化をするための暗号化関数を作用させて生成された出力値に、前記親機認証チップに設定された値で決まる第1の変換関数を作用させて得られる第1のレスポンス値を前記親機認証チップから受信することと；前記子機認証チップに秘密鍵として記憶された前記整数および前記乱数に対して暗号化をするための暗号化関数を作用させて生成された前記出力値に、前記子機認証チップに設定された値で決まる第2の変換関数を作用させて得られる第2のレスポンス値を前記子機認証チップから受信することと；前記第1のレスポンス値に、前記親機認証チップに設定された値と前記子機認証チップに設定された値の大きさの差の数で決まる第3の変換関数を作用させて得られる値と前記第2のレスポンス値を比較して、または前記第2のレスポンス値に前記第3の変換関数を作用させて得られる値と前記第1のレスポンス値を比較して、前記子機を認証すること、を含む。

[0008] また、本発明に従う認証システムは：乱数を生成する乱数生成手段と、前記乱数を前記親機認証チップおよび前記子機認証チップに送信する乱数送信手段と、前記親機認証チップおよび前記子機認証チップから、前記乱数に対する応答として第1のレスポンス値および第2のレスポンス値を受信するレスポンス値受信手段と、前記第1のレスポンス値および前記第2のレスポンス値を用いて、前記子機を認証する子機認証手段を含むプロセッサと；前記乱数を受信する第1の乱数受信手段と、秘密鍵としての整数を記憶する第1の記憶手段と、前記乱数および前記整数に暗号化のための暗号化関数を作用させて出力値を生成する第1の暗号演算手段と、前記出力値に、第1の設定された値で決まる第1の変換関数を作用させて前記第1のレスポンス値を生成する第1のレスポンス値生成手段と、を含み、前記プロセッサに接続されている親機認証チップと；前記乱数を受信する第2の乱数受信手段と、秘密鍵としての整数を記憶する第2の記憶手段と、前記乱数および前記整数に暗号化のための暗号化関数を作用させて出力値を生成する第2の暗号演算手段と、前記出力値に、第2の設定された値で決まる第2の変換関数を作用させ

て前記第2のレスポンス値を生成する第2のレスポンス値生成手段と、を含み、少なくとも前記プロセッサに接続されている子機認証チップと、を含み、前記プロセッサの前記子機認証手段は、前記第1のレスポンス値に、前記第1の設定された値と前記第2の設定された値の大きさの差の絶対値で決まる第3の変換関数を作用させて得られる数と前記第2のレスポンス値を比較して、または前記第2のレスポンス値に前記第3の変換関数を作用させて得られる数と前記第1のレスポンス値を比較して、前記子機を認証すること、を特徴とする。

[0009] また、本発明に従う認証チップは：整数である秘密鍵を記憶するメモリと；整数を受信する受信手段と；前記整数および前記秘密鍵に暗号化のための暗号化関数を作用させて生成された出力値に、変換関数を作用させて得られるレスポンス値を生成するレスポンス値生成手段と；前記レスポンス値を送信する送信手段、を含む。

発明の効果

[0010] 親機と子機に同一の値をチャレンジとして送信しても、レスポンスとして返ってくる値が異なるので、安全性が高く、安価に製造可能で、外部装置との通信機能が簡素な認証システムを得ることが出来る。

図面の簡単な説明

[0011] [図1] チャレンジアンドレスポンス認証プロトコルの概要である。

[図2] 共通鍵暗号を用いたチャレンジアンドレスポンス認証のプロトコルである。

[図3] 認証システムにおける、親機・子機側の実装形態の種類を示す図である。

[図4] 形態3をベースとした、通信トポロジの形態の種類を示す図である。

[図5] 形態3とトポロジ3の組み合わせ例（I2Cバス）を示す図である。

[図6] チャレンジアンドレスポンスプロトコルを示す図である。

[図7] 盗聴を利用した攻撃法を示す図である。

[図8] 改ざんを利用した攻撃法を示す図である。

[図9]本発明の第1の実施形態に従うシステムを示す図である。

[図10]本発明の第2の実施形態に従うシステムを示す図である。

[図11]本発明の第2の実施形態に従うシステムのCPUで実行される処理のフローチャートを示す図である。

[図12]本発明の第2の実施形態に従うシステムの認証チップで実行される処理のフローチャートを示す図である。

[図13]本発明の第3の実施形態に従うシステムを示す図である。

[図14]本発明の第3の実施形態に従うシステムのCPUで実行される処理のフローチャートを示す図である。

[図15]本発明の第3の実施形態に従うシステムの認証チップで実行される処理のフローチャートを示す図である。

[図16]本発明の第4の実施形態に従うシステムを示す図である。

[図17]本発明の第4の実施形態に従うシステムで、認証チップ0をCPUに装着する際にCPUで実行される処理のフローチャートを示す図である。

[図18]本発明の第4の実施形態に従うシステムで、認証チップ0をCPUに装着する際に認証チップ0で実行される処理のフローチャートを示す図である。

[図19]本発明の第4の実施形態に従うシステムで、認証チップiを含むi番目の子機を認証する際に、CPUで実行される処理のフローチャートを示す図である。

[図20]本発明の第4の実施形態に従うシステムで、認証チップiを含むi番目の子機を認証する際に、認証チップiで実行される処理のフローチャートを示す図である。

[図21]本発明の第5の実施形態に従うシステムを示す図である。

[図22]本発明の第5の実施形態に従うシステムのCPUで実行される処理のフローチャートを示す図である。

[図23]本発明の第5の実施形態に従うシステムの認証チップ0で実行される処理のフローチャートを示す図である。

[図24] 本発明の第6の実施形態に従うシステムを示す図である。

[図25] 本発明の第6の実施形態に従うシステムのCPUで実行される処理のフローチャートを示す図である。

[図26] 本発明の第6の実施形態に従うシステムの認証チップ0で実行される処理のフローチャートを示す図である。

[図27] 本発明の第7の実施形態に従うシステムを示す図である。

[図28] 本発明の第7の実施形態に従うシステムで、認証チップ0をCPUに装着する際にCPUで実行される処理のフローチャートを示す図である。

[図29] 本発明の第7の実施形態に従うシステムで、認証チップ0をCPUに装着する際に認証チップ0で実行される処理のフローチャートを示す図である。

[図30] 本発明の第7の実施形態に従うシステムで、認証チップiを含むi番目の子機を認証する際に、CPUで実行される処理のフローチャートを示す図である。

[図31] 本発明の第7の実施形態に従うシステムで、認証チップiを含むi番目の子機を認証する際に、認証チップiで実行される処理のフローチャートを示す図である。

発明を実施するための形態

[0012] 以下で図面を参照しながら、本発明に従う共通鍵を用いた認証システムを説明する。最初に比較例を説明するが、その後、本発明の第1から第7の実施形態を、比較例に対する利点を含めて説明する。尚、以下の説明において、複数の図面にわたり類似の構成要素または類似の機能を有する構成要素には、類似の参照符号を付して初出時以外、詳細な説明を省略する。

[0013] (比較例)

共通鍵暗号を用いて第一の装置が第二の装置を認証するシステムについて説明する。実施の形態における説明では、認証を行う側である第一の装置を「親機」、認証される側である第二の装置を「子機」と適宜称して説明する。親機はCPU100と親機認証チップ200を含み、子機は認証チップ3

00を含むシステムについて、秘密鍵が外部に漏洩しない、生産コストを削減する、通信トポロジをなるべく簡素にするという3つの制約の下で考えられるシステムについて説明する。

[0014] 認証チップにおいては、機器の正当性を確認するために、チャレンジアンドレスポンス認証プロトコルと呼ばれる通信プロトコルが利用されている。図1はチャレンジアンドレスポンス認証プロトコルの概要である。チャレンジアンドレスポンス認証とは、デジタル情報を用いた合言葉であり、認証を行う側（親機）から認証される側（子機）に対し、「チャンレジ」と呼ばれる乱数を送信する。これに対し、子機側では「チャンレンジ」に対して「レスポンス」と呼ばれる応答を生成し、親機に返信する。親機は、「チャンレンジ」に対する「レスポンス」の値を判断し、もし正しいならば、子機が正当な機器であると判断する。

[0015] 「チャンレンジ」として乱数を用いることで、対応する「レスポンス」が毎回変化する。これにより、再送攻撃に対する防止措置が実現される。再送攻撃とは、過去に外部から観察された応答を繰り返すことで、正当な機器の成りすましを行う攻撃である。つまり、乱数を用いない場合、チャンレジとレスポンスのペアが完全に固定値になるため、攻撃者がこのペアを観察することで、チャンレンジに対応する適切なレスポンスを知ることができ、このレスポンスを返すチップを製造することでチップ偽造が容易となるからである。例えるならば、「山」に対する「川」という合言葉しか用いないシステムであることを、悪意のある第三者が知ることができれば、常に「川」と応答することで成りすましが可能となる。

[0016] 「チャンレンジ」に対する「レスポンス」の生成は、暗号関数を用いた方法が一般的である。どの暗号関数を用いるかによってメリットとデメリットがあるが、認証チップの場合、回路規模の小ささを優先できるメリットがある共通鍵暗号を用いる方法が広く用いられている。

[0017] 図2は、共通鍵暗号を用いたチャレンジアンドレスポンス認証のプロトコルである。

図2に示すプロトコルでは、親機と子機の認証チップ間で、秘密鍵Kを事前共有している。これは、チップの製造時に秘密鍵Kの値を書き込むことで実現できる。この値が外部に漏洩しないことがセキュリティ上の重要な前提である。S100で、認証を行う側の親機は、乱数Cを生成し、認証される側の子機に送信する。この乱数Cは、「チャレンジ」と呼ばれる。S110で、子機は、受け取ったチャレンジCに対し、秘密鍵Kによる暗号化処理を実行し出力値 $R = \text{Enc}(C, K)$ を得る。そして、出力値Rを「レスポンス R_q 」として、親機に返信する。ただし、 $b = \text{Enc}(a, K)$ とは、共通鍵暗号を用いて、平文aを鍵Kにより暗号化した結果が暗号文bであることを表す。S120では、子機からのレスポンス R_q を受け取った親機は、子機と同様に、秘密鍵Kによる暗号化処理を実行し出力値 $R = \text{Enc}(C, K)$ を得て、レスポンスの正解値 $R_p = R$ を得る。子機からのレスポンス R_q が正解値 $R_p = R$ と一致するならば、親機は子機を正当な機器と認定する。チャレンジCに対応した正当なレスポンス R_p を生成できるのは、秘密鍵Kを有する場合に限られるので、親機は子機の正当性を確認することができる。

[0018] たとえば、図2に示されているような、チャレンジアンドレスポンスプロトコル自体は、理論上は安全な認証プロトコルである。しかし、現実世界の認証には制約があるため、安全にならない可能性が残される。特に、以下の3つの制約下での安全な認証を実現するのは困難であり、これらの制約下でも偽造品をふせぐことができる安全な認証プロトコル（システム）の実現が望まれている。

制約1：秘密鍵が外部に漏洩しないこと

制約2：生産コストの制約

制約3：通信トポロジの制約

[0019] 制約1の秘密鍵が外部に漏洩しないことは、耐タンパ性を有する認証チップを用いることで実現可能である。制約2の生産コストの制約については、認証チップの親機・子機側を同一のハードウェア構成とすることで、認証チップの生産コストを低く抑えることができる。制約3の通信トポロジの制約

は、低コスト向けの機器では、通信形態（トポロジ）をなるべく簡素なものとする必要があることに由来する。本発明に従うシステムでは、制約 1、2、3 の下でも、認証機器を偽造不可能な認証プロトコル（システム）を実現することができる。特に、制約 1 は、一般的に知られている耐タンパ技術を用いることで解決可能である。しかし制約 2 および 3 の下では、従来技術を用いただけでは安全な認証を実現することはできない。

[0020] 制約 2 により認証チップの実装形態が決定し、制約 3 によって通信トポロジが決定する。これらの決定された組み合わせが、認証の安全性を脅かす要因となることがある。最初に、図 3 を参照しながら、制約 2 により決定される認証チップの実装形態について説明する。

[0021] 認証システムにおける、親機、子機側の実装形態の種類を図 3 に示す。認証システムの基本要件は以下の通りである。

一親機と子機から構成される。

一親機と子機それぞれに、認証チップまたは CPU が装着される。ただし、親機側には全体の機能管理を行うためのコントローラとしての CPU が必ず装着される。

一親機と子機に装着された認証チップもしくは CPU が通信を行い、認証プロトコルを実行することで、子機が正規品であるかどうかを親機側が判定する。

一子機が正規品であると親機側が判定した場合、親機側の CPU が子機のリソースを親機のために利用することを許可する。例としては、プリンタカートリッジの場合、親機のプリンタが子機のインク使用を許可すること、医療機器の場合、親機の機器が子機のオプション機器使用を許可することを含む。このように、チャレンジアンドレスポンスプロトコルでは、不正を防ぐためには、鍵の値が外部に漏れないことが前提であり、固定鍵による暗号化、復号処理は、サイドチャンネル攻撃で狙われる可能性がある。

[0022] このシステムの実装形態として、図 3 に示されるように、以下の形態 1 ～ 3 が考えられる。

形態１：親機側をＣＰＵ１００、子機側を認証チップ３００により実装する。

形態２：親機側を、認証チップ内蔵型のＣＰＵ１００'、子機側を認証チップ３００により実装する。

形態３：親機側をＣＰＵ１００と認証チップ２００の組み合わせ、子機側を認証チップ３００により実装する。

[0023] 形態１では、親機はＣＰＵ１００を含み、子機は認証チップ３００を含む。子機に含まれる認証チップ３００は、通信制御部３０１、暗号回路３０２および秘密鍵を記憶するメモリ３０３を含む。通信制御部３０１はプロセッサ外部との通信を処理し、プロセッサとして実装されても良い。暗号回路３０２は入力（たとえばチャレンジ）に対して、暗号化関数を作用させて、レスポンス（応答）として出力値を生成するもので、専用の回路であっても、汎用コンピュータとして実装されても良い。秘密鍵は暗号回路３０２から読み出され、メモリ３０３に記憶され得るが、メモリ３０３は不揮発性であることが好ましい。認証プロトコルとして共通鍵暗号を用いた場合、親機のＣＰＵ１００に秘密鍵を書き込む必要がある。この秘密鍵の値は、すべての親機、子機を通じて製品に共通の鍵であるため、耐タンパ性がないＣＰＵ１００から秘密鍵が漏洩する可能性が高く、偽造が容易となる。すなわち、制約１が満たされない。

[0024] 形態２では、親機は認証チップ２００'を含む認証チップ内蔵型ＣＰＵ１００'を含む。認証チップ２００'は、ＣＰＵに内蔵されている点で、以下で述べる認証チップ２００と異なるが構成は同一である。すなわち、認証チップ２００'は、認証チップ３００の通信制御部３０１、暗号回路３０２および秘密鍵を記憶するメモリ３０３と同一である通信制御部２０１、暗号回路２０２および秘密鍵を記憶するメモリ２０３を含む。子機は形態１と同様である。本形態では、耐タンパ機能を備えた認証チップ２００'をＣＰＵ１００'に内蔵することで、ＣＰＵからの秘密鍵漏洩リスクが消えるため、制約１を満たす。しかし、親機側が認証チップ内蔵型のＣＰＵ１００'、子機

側が通常の認証チップ300であるため、制約2が満たされない。

[0025] 形態3では、親機はCPU100および認証チップ200を含み、子機は認証チップ300を含む。本実施形態では、親機、子機ともに耐タンパ機能を備えた認証チップ200、300を用いるため、制約1を満たす。さらに、親機、子機共に、同一の認証チップを用いるため、制約2を満たす。

[0026] そこで、形態3をベースとした、通信トポロジの形態の種類を図4に示す。

トポロジ1のブリッジ型は、親機側の認証チップから、親機CPUと子機認証チップそれぞれに対し別々に通信線を結ぶ形式である。親機認証チップに2つの通信ポートが必要とされる（同じハードウェア構成である子機認証チップにも2つの通信ポートが必要）ため、低コストに向かない。すなわち、制約3が満たされない。

[0027] トポロジ2のハブ型は、親機に含まれるCPU100から、親機の認証チップ200と子機の認証チップ300それぞれに対し別々に通信線を結ぶ形式である。CPUに2つの通信ポートが必要とされるため、低コストに向かない。すなわち、制約3が満たされない。

[0028] トポロジ3のシリアルバス型は、親機に含まれるCPU100、親機の認証チップ200および子機の認証チップ300の全てを、共有する通信線でバス状に接続する形式である。親機のCPU100、親機の認証チップ200、および子機の認証チップ300の全ての通信ポートが1つで良いため、低コストに向く。すなわち、制約3を満たす。

[0029] すなわち、形態3をベースにシリアルバス型の接続を行うのが、上記制約1、2、3を満たすための好適な形態である。

しかし、これらの制約を満たしても、以下のようなセキュリティ上の問題が発生することがある。

[0030] 図5は、形態3とトポロジ3の組み合わせ例を示す図である。シリアルバスI/Fとしては、多様な種類が知られているが、図5ではI2Cバス400によって、CPU101、親機の認証チップ200、および子機の認証チ

チップ300がバス接続されている。I2Cバス400は、データ線420とクロック線440の2線により、全ての機器が接続されている。I2Cバス400は、2線の通信線1本のみを用いて複数の機器を接続可能、というメリットがある反面、全機器が一本の線でつながっているため、データの盗聴や改ざんが容易という問題があることが知られている。

[0031] 図6～8を用いてセキュリティ上の問題について説明する。

I2Cバス上のデータの盗聴や改ざんが容易である、という性質を利用すると、図5に示した通信トポロジ上で、図2に示したチャレンジアドレスポンスプロトコルをそのまま実行すると、図6に示されるような状態となる。

[0032] 図6では、CPU100が親機の認証チップ200および子機の認証チップ300それぞれにバス400を介してチャレンジCを送信し、子機の認証チップ300からのレスポンスR_Qと、親機の認証チップ200からのレスポンスR_Pを比較し、R_PとR_Qが一致したならば、CPU100は子機の認証チップ300を含む子機が正規品であると判断する。CPU100がチャレンジCを送信する順序を、子機の認証チップ300を含む子機から開始するか、親機の認証チップ200を含む親機から開始するかに応じて、2通りの方法が存在する。

[0033] 親機に先にチャレンジCを送信する場合、CPU100は、バス400を介して、親機の認証チップ200にチャレンジCを出力する。親機の認証チップ200は、通信制御部201、暗号回路202、鍵203を含み、CPU100から出力されたチャレンジCは、通信制御部201で受信される。その後、チャレンジCは、暗号回路202に入力され、暗号回路202では、鍵203を用いて、レスポンスR_Pを得る。このレスポンスR_Pはバス400を介して、CPU100に送られる。次に、CPU100は、バス400を介して、子機の認証チップ300にチャレンジCを出力する。チャレンジCは、子機の認証チップ300の通信制御部301で受信され、その後、子機の認証チップ300の暗号回路302に入力される。暗号回路302では

、チャレンジ及び鍵303と暗号化関数を用いて、レスポンス R_Q を得る。このレスポンス R_Q は、バス400を介して、CPU100に送られる。親機からのレスポンス R_P と、子機からのレスポンス R_Q を受け取ったCPU100は、両者を比較し、一致すれば子機は正規品と判定する。

[0034] 子機に先にチャレンジCを送信する場合は、CPUはまず、子機の認証チップ300にチャレンジCを出力し、子機の認証チップ300からレスポンス R_Q を得たのち、親機の認証チップ200にチャレンジCを出力し、親機の認証チップ200からレスポンス R_P を得る。その後、両者を比較し、一致すれば子機は正規品と判定する。

[0035] 子機が正規品である限りは、子機の認証チップ300を含む子機から認証を開始しても、親機の認証チップ200を含む親機から認証を開始しても問題はない。

しかし、発明者独自の分析により、図6のプロトコルの安全性評価を行った結果、上の2通りの方法に対して異なる攻撃法1、2で、I2Cバス400上の通信データの盗聴・改ざんを利用することで、秘密鍵を知らなくとも偽造品の製造が可能であることが判明した。以下に、図7、8を参照しながら、攻撃法1、2を説明する。

[0036] (攻撃法1)

図7は、図6のプロトコルに対して、I2Cバス400上の盗聴を用いた場合の攻撃法である(以下、攻撃法1と呼ぶ)。攻撃対象は、親機の認証チップ200を含む親機、子機の認証チップ300を含む子機の順にレスポンスを生成する場合のプロトコルである。子機の認証チップ300が正規品の場合、CPU100は親機、子機からのレスポンス R_P 、 R_Q をこの順番で受信し、正規品の判定を行う。これに対し、偽造品である子機の認証チップは、次のように振舞う。偽造認証チップ300'は、I2Cバス400を通じて、親機がCPU100に返信するレスポンス R_P を観察し、偽造認証チップ300'の(図示されていない)内部レジスタに格納する。次に、CPUから子機へのチャレンジCが送信されるので、これに対して盗聴により観察され

た R_P をそのまま子機のレスポンス R_Q として返す。こうすることで、CPUは同一のレスポンスを受け取るため、 R_P と R_Q が一致し、CPU100は偽造認証チップを正規品と判定してしまう。このように攻撃法1では、偽造認証チップは、I2Cバス上のデータを単純に盗聴すればよいから、秘密鍵の値を知らなくとも攻撃が成功する。

[0037] (攻撃法2)

図8は、図6のプロトコルに対して、改ざんを用いた場合の攻撃法である(以下、攻撃法2と呼ぶ)。攻撃対象は、子機、親機の順にレスポンスを生成する場合のプロトコルである。子機の認証チップ300が正規品の場合、CPU100は子機、親機からのレスポンス R_Q 、 R_P をこの順番で受信し、正規品の判定を行う。これに対し、偽造品である子機の認証チップ300'は、次のように振舞う。偽造認証チップ300'は、適当な値 X を生成し、CPU100からのチャレンジ C に対するレスポンス R_Q として返信する。この X はどんな値でも良い。次に、認証チップ300'を含む子機はI2Cバス400上のデータを盗聴し、CPU100から親機の認証チップ200を含む親機へのチャレンジ C が入力されたならば、先ほど生成した値 X を親機認証チップ200の代わりに返信する(成りすまし)。この成りすましは、偽造認証チップ側の通信処理タイミングが非常にシビアであるが、原理的には可能である。このタイミング制御のシビアさを攻撃者が回避する手法としては、CPU100と親機の認証チップ200の間の通信線400を切断する方法が挙げられる。親機側の回路の物理改変が必要となるが、この攻撃法2を用いることで、子機によるなり済ましを容易にすることができ、偽造品を正規品と誤認させることができる。

[0038] 上述のように、比較例では、親機からのレスポンス R_P と子機からのレスポンス R_Q が同じであるかどうかで認証の成否を判断しており、盗聴をする攻撃法1、成りすましをする攻撃法2等を用いることによって、認証の安全性が脅かされるおそれがある。

以下では、上記制約1、2、3を満たす形態3をベースにシリアルバス型

の接続を行ったシステムであって、認証の安全性が改善されたシステムについて説明する。

[0039] (第1実施形態)

図9を参照して、本発明の第1実施形態に従う共通鍵暗号を用いた認証システムを説明する。本実施形態は本発明の基本的実施形態でもある。本実施形態では、制約1、2、3を満たす図6のプロトコルは、上記攻撃法1、攻撃法2に脆弱である、という問題に対し、攻撃法1、2に対しても安全な認証を実現するコンパクトなシステムを実現することができる。

[0040] 本発明の第1実施形態に従うシステムを図9に示す。上記比較例では、親機認証チップ、子機認証チップともに、同一のレスポンスを返信していたため、盗聴や成りすましによる攻撃が容易であった。これに対し、本実施形態では、チャレンジに対してレスポンスを生成するレスポンス生成部を認証チップ内で複数準備し、いずれのレスポンス生成部を用いるかを、親機および子機に含まれる認証チップ内部の不揮発性メモリに格納されている情報に基づいて選択することを特徴とする。この情報は、同一ハードウェア構成を持つ親機および子機の認証チップが、自分自身が親機側か、子機側の認証チップであるかを識別するために用いられる情報であり、一般的にはI2Cバスのアドレス情報であって良い。すなわち、I2Cアドレスにより、認証チップが親機であるか、子機であるかが決定される。このアドレス情報が子機側であった場合、子機は通常のレスポンス（出力値と呼ぶ） $R_Q = E_{nc}(C, K)$ を返信する。これに対し、アドレス情報が親機側であった場合、親機側は通常のレスポンス、すなわち出力値を関数Fで変形させたレスポンス $R_P = F(E_{nc}(C, K))$ を生成する。ただし、関数Fは外部に対して非公開の関数である。これらのレスポンス R_P 、 R_Q を受け取ったCPUは R_Q から計算された $F(R_Q)$ が R_P に等しいか判定を行い、もし等しいならば子機が正規品であると判断する。正規品によるレスポンスならば、 $F(R_Q) = F(E_{nc}(C, K)) = R_P$ であり等しい値となる。外部非公開の関数Fによりレスポンスが変形されているため、攻撃1、攻撃2に示したような方法を用い

て、親機および子機のレスポンス値 R_p および R_q が同一とするような、CPUに対して偽装する攻撃は通用しないため、安全な認証を実現することができる。

[0041] 図9に示されているように、本実施形態に従うシステムは、親機によって子機を認証するシステムであって、親機に含まれるCPU100および親機の認証チップ200、子機に含まれる子機の認証チップ300、ならびにCPU100、親機の認証チップ200および子機の認証チップ300を接続するバス400を含んでいる。CPU100、親機の認証チップ200および子機の認証チップ300はバス400を介してシリアルバス接続されている。バスインターフェイス(I/F)としては、I2Cバスを例示することができる。

[0042] CPU100は、乱数であるチャレンジCを生成し、親機の認証チップ200および子機の認証チップ300に出力する。乱数は整数であることが望ましい。また、CPU100は、親機からのレスポンス R_p と子機からのレスポンス R_q を受ける。そして、CPU100は、子機からのレスポンス R_q に関数Fを作用させた $F(R_q)$ とレスポンス R_p を比較して、子機の認証を行う。すなわち、CPU100は、親機からのレスポンス R_p と子機からのレスポンス R_q の差を補償する関数Fを記憶している。本実施形態では、親機の認証チップ200でレスポンスRに作用される関数Fと、CPUで子機からのレスポンス R_q に作用される関数Fは同一の関数である。

乱数はCPU100に含まれる乱数生成手段で生成されても良い。乱数は整数であることが好ましい。チャレンジCの外部への出力(送信)は乱数送信手段によって行われても良い。レスポンスを受けること(受信)は、レスポンス値受信手段によって行われても良い。さらにCPU100は、親機からのレスポンス R_p と子機からのレスポンス R_q を比較し、子機が正規品かどうか判定する子機認証手段を含む。

[0043] 親機の認証チップ200および子機の認証チップ300は、上記制約2の観点からは同一の構造を有することが好ましい。認証チップ200、300

はそれぞれ、第1のレスポンス生成部220a、320a、第2のレスポンス生成部220b、320b、レスポンス選択部230、アドレス情報保持部240、340を含む。

[0044] 第1のレスポンス生成部220a、320aは、暗号化関数を記憶し、従来のようにCPU100からの入力（チャレンジC）および鍵に暗号化関数を作用させてレスポンスとして出力値Rを生成する。暗号化関数は公知のもので良い。この第1のレスポンス生成部は、暗号演算手段とも呼ばれる。第2のレスポンス生成部220b、320bは、関数Fを記憶しており、CPU100からの入力（チャレンジC）および鍵に暗号化関数を作用させて得られた出力値Rに、さらに関数Fを作用させて、出力値Rとは異なる数F（R）を生成する。この第2のレスポンス生成部は、第1のレスポンス値生成手段とも呼ばれる。アドレス情報保持部240、340は個々の認証チップに固有のアドレスを記憶する不揮発性メモリであっても良い。親機の認証チップ200および子機の認証チップ300は、上記の構成要素のほかに、たとえばCPU100などの外部装置からの信号を受信したり、外部装置に信号を送信したりする（図示されていない）通信手段を含む。通信手段の例には、図10に示されているデータ入出力インターフェイス（I/F）260、360が含まれる。データ入出力インターフェイス（I/F）は、プロセッサ100で生成された乱数を受信する受信手段、またはレスポンス生成部で生成されたレスポンスをプロセッサ100に送信する送信手段として機能する。

[0045] 本実施形態では、鍵は共通鍵（秘密鍵とも呼ばれる）である。つまり、第1のレスポンス生成部220a、320aでは共通鍵暗号のアルゴリズムが実行される。レスポンス選択部230、330では、アドレス情報保持部240、340に記憶されたシリアルバスのアドレス値に応じてレスポンス生成部を選択する。第1のレスポンス生成手段と第2のレスポンス生成手段とでレスポンス値生成手段が形成される。

[0046] 上記では、子機の認証チップ300は、CPU100から入力されるチャ

レンジCに対して、通常のレスポンス $R_Q = \text{Enc}(C, K)$ を返すものとした。しかし、子機の認証チップ300も、 $\text{Enc}(C, K)$ にある関数Hを作用させた $H(\text{Enc}(C, K))$ を返しても良い。この場合でも、CPU100が、親機からのレスポンス R_P と子機からのレスポンス R_Q の差を補償する関数を記憶していれば、上記の認証方法は機能する。レスポンス R_P と R_Q の差を補償する関数は、親機の認証チップ200のメモリ240に記憶されているアドレス情報と子機の認証チップ300のメモリ340に記憶されているアドレス情報から一意に決まることが好ましい。メモリ240、340はアドレス記憶手段を構成する。

[0047] 親機の認証チップ200および子機の認証チップ300は同一の構成であるが、親機の認証チップ200として用いられるものでは、第2のレスポンス生成部220bが、子機の認証チップ300として用いられるものでは、第1のレスポンス生成部320aが機能するように、出荷時に設定されてもよい。このように出荷時に設定されることによって、認証チップとしては同一のものを大量生産することができること、出荷時に設定するので高い安全性を確保できること、などの利点を得ることができる。

[0048] このような構成を採用することによって、認証システム10は、仮にバス400がI2Cバスであっても、親機の認証チップ200からのレスポンスと子機の認証チップ300からのレスポンスが異なるため、認証の安全性を高めることができる。

[0049] また、本実施形態では、親機の認証チップ200および子機の認証チップ300は同一の構成を有している。また、CPU100、親機の認証チップ200および子機の認証チップ300はI2Cバス接続されている。このような構成を採用することによって、各認証チップ200、300が外部との通信のために備える通信ポートは1つでよく、認証チップは、製造コストを抑えつつも、小型に製造することができる。

また、各認証チップにおいて、変換関数は外部に対して非公開、即ち不揮発性メモリに記憶されている。従って、高い安全性を得ることができる。

[0050] (第2実施形態)

図10から12を参照しながら、本発明の第2実施形態を説明する。

[0051] 本実施形態に従うシステム10では、親機に対して、最大 n 個の子機が接続されている。親機はCPU100と認証チップ(認証チップ0)200を備え、最大 n 個の子機はそれぞれ1個ずつの認証チップ(認証チップ1～ n)300__1、...、300__ n を備える。 n は任意の自然数であり、用途によっては子機側に複数の認証チップが接続される場合がある。例えば、インクカートリッジを4色用いる場合、子機はそれぞれの色に対して1個ずつの認証チップを有し、計4個の認証チップを備えることになる。それぞれの認証チップ300__1、...、300__ n は、同一ハードウェア構成を持ち、各チップ内部の不揮発性メモリ326にはシリアルバスのアドレス値が書かれている。このアドレス値は、シリアルバス上の通信エンティティを区別するために用いられる値であり、CPU100、認証チップ300__1、...、300__ n に対してユニークな値が与えられる。このアドレスのユニーク性は、図13のシリアルバス400に接続されている通信エンティティ間で保たれればよく、流通している全ての認証チップ固体間で異なる値である必要はない。

[0052] 親機のCPU100は、 i 番目の子機が正規品であるかの確認を行うために、認証チップ0および認証チップ i それぞれに対してチャレンジ C を送信し、認証チップ i からのレスポンス R_i と、認証チップ0からのレスポンス R_0 の比較を行う。それぞれの認証チップは、 $n+1$ 個のデータ変換手段を備える。それぞれの認証チップは、受信したチャレンジ C からメモリ226に格納されている秘密鍵を用いてレスポンス R を生成するレスポンス生成回路224と、変換0～ n の実行を行う演算部220 a ～ n 、変換関数のパラメータとしてのアドレス情報を記憶するメモリ240およびメモリ240に記憶されたアドレス情報を元に演算部220 a ～ n から一つを選択するセレクタ223を含む。演算部220 a ～ n はそれぞれ、変換0～ n の変換関数を記憶している。

[0053] i 番目の認証チップが選択するデータ変換手段への入力を X 、出力を Y とすると、 X と Y の関係の例として、関数 F を用いて $Y = F^{t_i}(X)$ を例示することができる。ただし、 $Y = F^z(X)$ は、 X を関数 F に対して z 回連続適用した結果である。例えば $F(X) = X + 1$ の場合、 $F^1(10) = F(10) = 10 + 1 = 11$ 、 $F^2(10) = F(F(10)) = 12$ 、 $F^4(10) = 14$ 、 $F^0(10) = 10$ 、である。 t_i はメモリ 240 に記憶されているアドレス情報から決まる定数であり、 i 番目の認証チップが用いる変換手段に対応する関数 F の繰り返し適用回数を表す。関数 $F(X)$ は外部に対して非公開の関数である 1 入力 1 出力関数ならばどのような関数でもよいが、ハードウェア実装を考慮すると、なるべく回路規模が小さいものであることが良く、例えば、以下の処理が望ましい。

(1) フィードバックシフトレジスタ (LFSR) によるシフト処理を定数ビット分繰り返す

(2) 左 1 ビットのシフト後、 2^m 個の要素を有する拡大ガロア体上 $GF(2^m)$ 上の特性多項式の剰余演算処理を定数ビット分繰り返す (mod)

(3) ハッシュ (Hash) 関数

[0054] 上記 (1) の例としては、

$$F(X) = \text{LFSR}(X, 128),$$

ただし $\text{LFSR}(X, 1)$ は、 X を LFSR (線形フィードバックシフトレジスタ) 処理により 1 ビットだけシフトした値、が含まれる。

[0055] 上記 (2) の例としては、

$$F(X) = X^{64} \bmod q(X),$$

ただし $q(X)$ は 128 ビット既約多項式、 mod は剰余、を表す。

[0056] 上記 (3) の例としては、

$$F(X) = \text{SHA1}(X) \text{ の最下位 } 128 \text{ ビット},$$

ただし $\text{SHA1}(X)$ は X の $\text{SHA}-1$ ハッシュ関数出力、である。

[0057] もちろん、(1) ~ (3) の具体例は、上記関数に限定されない。

変換手段の差異は、同一関数を繰り返す回数のみに関する差異であるため

、認証チップでは関数Fを処理する回路のみ実装すればよく、 $n+1$ 個の変換手段それぞれに対して異なる回路を実装する必要は無いため、製造コストを低く抑えることができる。

[0058] 認証チップ0と認証チップiからの応答 R_P と R_Q を受け取った親機CPU100は、これらのレスポンス比較を以下により行う。

$t_i > t_0$ の場合： $F^{t_i-t_0}(R_P) = R_Q$ なら正規品、そうでなければ偽造品と判定され、 $t_i < t_0$ の場合： $R_P = F^{t_0-t_i}(R_Q)$ なら正規品、そうでなければ偽造品と判定される。

[0059] 上記の比較が正しい理由は、正規品の場合、 $R_P = F^{t_0}(R)$ 、 $R_Q = F^{t_i}(R)$ であり、もし $t_i > t_0$ ならば、 $F^{t_i-t_0}(R_P) = F^{t_i-t_0}(F^{t_0}(R)) = F^{t_i}(R)$ が R_Q と一致するからであり、もし $t_i < t_0$ ならば、 $F^{t_0-t_i}(R_Q) = F^{t_0-t_i}(F^{t_i}(R)) = F^{t_0}(R)$ が R_P と一致するからである。

[0060] 図10は本発明の第2の実施形態に従うシステム10を示す図である。本システム10は、図9に示されたシステムと同様、親機に含まれるCPU100および親機の認証チップ200、子機に含まれる子機の認証チップ300__1、...、300__n、ならびにCPU100、親機の認証チップ200および子機の認証チップ300を接続するバス400を含んでいる。CPU100、親機の認証チップ200および子機の認証チップ300はバス400を介してI2Cバスシリアルバス接続されている。

[0061] 親機に含まれる0番目の認証チップ200（以下、認証チップ0としても参照する）は、バス400を介してCPU100からの信号を受信したり、CPU100に信号を送信したりするデータ入出力インターフェイス（I/F）260、データ入出力I/F260で受信したチャレンジCに暗号化関数を作用させて、レスポンスとしての出力値Rを生成するレスポンス生成回路224、レスポンス生成回路224で生成された出力値Rに変換0～nを作用させる演算部222a～n、変換関数のパラメータとしてのアドレス情報を記憶するメモリ240およびメモリ240に記憶されたアドレス情報を

元に演算部 222a~n から一つを選択するセレクタ 223 を含む。セレクタ 223 はレスポンス選択部に相当する。レスポンス生成回路 224 には秘密鍵が記憶されているメモリ 226 が含まれている。図 9 のシステムは、本システムで子機の数 が 1 つの場合であり、図 9 の第 1 のレスポンス生成部 220a は、レスポンス生成回路 224、変換 0 を行う演算部 222a、およびセレクタ 223 に相当する。演算部 220a~n、セレクタ 223、レスポンス生成回路 224、アドレス情報を記憶するメモリ 240 からレスポンス値生成手段が形成される。

[0062] 子機の認証チップ 300_1、...、300_n も親機の認証チップ 200 と同様の構成をしている。すなわち、バス 400 を介して CPU 100 からの信号を受信したり、CPU 100 に信号を送信したりするデータ入出力インターフェイス (I/F) 360、データ入出力 I/F 360 で受信したチャレンジ C に暗号化関数を作用させて、レスポンスとしての出力値 R を生成するレスポンス生成回路 324、レスポンス R を生成するレスポンス生成回路 324 で生成されたレスポンスに変換 0~n を作用させる演算部 322a~n、変換関数のパラメータとしてのアドレス情報を記憶するメモリ 340 およびメモリ 340 に記憶されたアドレス情報を元に演算部 322a~n から一つを選択するセレクタ 223 を含む。レスポンス生成回路 324 には秘密鍵が記憶されているメモリ 326 が含まれている。図 9 の第 2 のレスポンス生成部 320b は、レスポンス生成回路 324、変換 1 を実行する演算部 322b、およびセレクタ 323 に相当する。演算部 320a~n はそれぞれ、変換 0~n の変換関数 F を記憶している。演算部 320a~n、セレクタ 323、レスポンス生成回路 324、アドレス情報を記憶するメモリ 340 からレスポンス値生成手段が形成される。

[0063] セレクタ 223、323 は、それぞれメモリ 240、340 に記憶されたシリアルバスのアドレス値に応じて変換関数を選択するレスポンス選択部としても機能する。

バス 400 は本実施形態では I2C バスである。I2C バスを用いること

によって、CPUと認証チップの接続は簡素なものとしことができ、システム全体のコストを抑えることができる。

[0064] 図11は、本発明の第2の実施形態に従うシステム10のCPU100で実行される処理のフローチャートを示す図であり、図12は本発明の第2の実施形態に従うシステムの認証チップで実行される処理のフローチャートを示す図である。

[0065] CPU100は、S1101で認証対象の子機番号*i*を決定し、S1102に進む。S1102では、チャレンジCとして用いるために乱数を生成する。S1102の次のS1103では、S1102で生成されたチャレンジCをS1101で決定された*i*番目の子機の認証チップ300__*i*に送信する。

[0066] *i*番目の子機の認証チップ300__*i*では、S1151で親機のCPU100によって送信されたチャレンジCを受信する。次に、S1152では、認証チップ300__*i*のレスポンス生成回路324中の秘密鍵記憶部326に記憶されている秘密鍵KとチャレンジCを用いて、レスポンスとしての出力値 $R = \text{Enc}(C, K)$ を生成する。次のS1153では、内部不揮発性メモリ340に記憶されたアドレス情報を元に、変換の繰り返し回数 t_i を決定する。繰り返し回数 t_i は、たとえば、アドレスに現れる数を順に並べて作られる整数であっても良い。S1153に引き続くS1154からS1157では、レスポンスRに t_i 回関数Fを作用させる。より詳細には、S1154で整数であるダミー変数*j*に0を代入する。S1155でダミー変数の値を1つ増加させ、S1156で*j*と t_i を比較する。S1156で*j*が t_i より小さければS1157で現在のレスポンス値に関数Fを作用させ、レスポンス値を更新して、S1154に戻る。S1156で*j*が t_i より大きければ現在のレスポンスRの値をCPU100に送信して、*i*番目の子機の認証チップ300__*i*の処理を終了する。

[0067] CPU100は、S1104で*i*番目の子機の認証チップ300__*i*からのレスポンスRの値をレスポンスR₀として受信する。次に、S1105でチ

チャレンジCを今度は、親機の認証チップ200に送信する。

- [0068] 親機の認証チップ200では、先のi番目の子機の認証チップ300__iと同じ処理を行うが、i番目の子機の認証チップ300__iの場合とは、関数Fを作用させる回数が t_0 であり、 t_i とは異なる。
- [0069] CPU100は、S1106で親機の認証チップ200からのレスポンスRの値をレスポンス R_p として受信する。次に、S1107で t_i と t_0 を比較し、 t_i が t_0 より大きければS1108に進み、そうでなければS1109に進む。S1108では、 R_p に関数Fを、 $t_i - t_0$ 回作用させた値と R_0 が等しいかどうかを判定する。もし等しければi番目の子機を正規品と判定してCPU100の処理を終了する。そうでなければ、i番目の子機を正規品ではないと判定する。S1109では、 R_0 に関数Fを、 $t_0 - t_i$ 回作用させた値と R_p が等しいかどうかを判定してCPU100の処理を終了する。もし等しければi番目の子機を正規品と判定してCPU100の処理を終了する。そうでなければ、i番目の子機を正規品ではないと判定してCPU100の処理を終了する。
- [0070] このような構成を採用することによって、本実施形態のシステム10では、レスポンス R_p 、 R_0 が異なるため、攻撃法1、攻撃法2に示したような、盗聴及び改ざんによる攻撃を防ぐことができる。
- [0071] (第3実施形態)
- 図13～15を参照して本実施形態に従う認証システム20を説明する。図13は、本実施形態のシステム20を示す図、図14は、本実施形態に従うシステムのCPU1100で実行される処理のフローチャートを示す図であり、図15は本実施形態に従うシステム20の認証チップ1200、1300__iで実行される処理のフローチャートを示す図である。
- [0072] 実施形態2に従うシステム10では、関数Fは非公開関数であっても、ハードウェア上で効率よい処理ができるFの種類は非常に限定的であるため、データ盗聴の結果から正しいFの内容が特定される可能性はなくはない。関数Fの内容が攻撃者に特定されたならば、攻撃1、攻撃2と同様の考え方に

基づきつつ、関数Fにより変形されたレスポンスを返すように変更した攻撃法を用いることで、秘密鍵の値を知らなくとも偽造品を製造することが可能となる。本実施形態では、通信データの盗聴から変換手段を特定することがより困難なシステム20が提供される。

[0073] 本実施形態に従う子機認証システム20は、基本的には、第2実施形態のシステム10と同じであるが、事前共有された定数S（共有値S）が書き込まれている不揮発性メモリ1280、1380を認証チップ1300__1～1300__n（認証チップ1～認証チップn）が含んでいる。この定数Sは、CPU1100および認証チップ1200、1300__1～1300__nの製造時に書き込まれる値である。i番目の認証チップが選択するデータ変換手段への入力をX、出力をYとすると、XとYの関係は、関数 $F_{s,c}$ を用いて $Y = F_{s,c}^{(i)}(X)$ と表すことができる。ただし、 $F_{s,c}(X)$ は定数S、チャレンジCをパラメータとした1入力1出力関数を表し、例えば、以下のような演算が挙げられる。

- (4) パラメータ付XOR加算
- (5) パラメータ付剰余演算
- (6) パラメータハッシュ演算
- (7) パラメータ付LFSR-XOR減算

[0074] 上記(4)の例としては

[数1]

$$F_{s,c}(X) = (S + X) \oplus C、$$

が含まれる。

[0075] 上記(5)の例としては、||をビット結合、 $q(X)$ を128ビット既約多項式、modを剰余として、

$$F_{s,c}(X) = (S || C || X) \bmod q(X)、$$

が含まれる。

[0076] 上記(6)の例としては、

[数2]

$$F_{s, c}(X) = \text{SHA1}(S \oplus C \oplus X),$$

の最下位128ビット、が含まれる。

上記(7)の例としては、

[数3]

$$F_{s, c}(X) = \text{LFSR}(S, 19) \oplus \text{LFSR}(C, 42) - \text{LFSR}(X, 21),$$

が含まれる。

[0077] (4)～(7)の具体例は、上記関数に限定されない。

変換手段がチャレンジCと、外部から観察できない定数Sの影響を受けるため、変換のバリエーションが大きく増える。これにより、前述の通信データ盗聴を用いた変換手段の解読が攻撃者にとって困難となる。例えば、Sを128ビットパラメータとすることで、攻撃者の総当たり総数は 2^{128} となり、現実時間内では解読不可能となる。すなわち、本実施形態のシステム20を用いることで、通信データ盗聴による変換手段の解読を用いた攻撃を回避することができる。また、本実施形態に従うシステム20では、攻撃者がCPU1100に対して高度なりバースエンジニアリングを行い、関数Fの処理内容を解読した場合にも安全性を確保することができる。

[0078] 本実施形態のシステム20のCPU1100は、共有値を記憶するメモリ1140を含んでいる点を除けば、前実施形態のCPU100と同一である。

親機の認証チップ1200は、実施形態2の親機の認証チップ200と類似の構成を有しているが、共有値Sを記憶するメモリ1280を含む点が異なっている。

[0079] 子機の認証チップ1300__1、...、1300__nは、実施形態2の子機の認証チップ認証チップ300と類似の構成を有するが、共有値Sを記憶するメモリ1380を含む点が異なっている。また、子機の認証チップ1300__1、...、1300__nは、親機の認証チップ1200と同じ構成を有する。

[0080] 図14は、本発明の第3実施形態に従うシステムのCPU1100で実行される処理のフローチャートを示す図であり、図15は本発明の第3の実施形態に従うシステムの認証チップで実行される処理のフローチャートを示す図である。

[0081] CPU1100は、S1301で認証対象の子機番号*i*を決定し、S1302に進む。S1302では、チャレンジCとして用いるために乱数を生成する。S1302の次のS1303では、S1302で生成されたチャレンジCをS1301で決定された*i*番目の子機の認証チップ1300__*i*に送信する。

[0082] *i*番目の子機の認証チップ1300__*i*では、S1351で親機のCPU1100によって送信されたチャレンジCを受信する。次に、S1352では、認証チップ1300__*i*のレスポンス生成回路324中の秘密鍵記憶部326に記憶されている秘密鍵KとチャレンジCを用いて、レスポンスとしての出力値 $R = \text{Enc}(C, K)$ を生成する。次のS1353では、内部不揮発性メモリ340に記憶されたアドレス情報を元に、変換の繰り返し回数 t_i を決定する。実施形態2と同様、繰り返し回数 t_i は、たとえば、アドレスに現れる数を順に並べて得られる整数であっても良い。S1353に引き続くS1354からS1357では、出力値Rに t_i 回関数 $F_{s,c}$ を作用させる。関数 $F_{s,c}$ は、共有値SおよびチャレンジCによって決まる。より詳細には、S1354で整数であるダミー変数*j*に0を代入する。S1355でダミー変数の値を1つ増加させ、S1356で*j*と t_i を比較する。S1356で*j*が t_i より小さければS1357で現在のレスポンス値に関数 $F_{s,c}$ を作用させ、レスポンス値を更新して、S1354に戻る。S1356で*j*が t_i より大きければ現在のレスポンスRの値をCPU1100に送信して、*i*番目の子機の認証チップ1300__*i*の処理を終了する。

[0083] CPU1100は、S1304で*i*番目の子機の認証チップ1300__*i*からのレスポンスRの値をレスポンス R_o として受信する。次に、S1305でチャレンジCを今度は、親機の認証チップ1200に送信する。

- [0084] 親機の認証チップ1200では、先の i 番目の子機の認証チップ1300__ i と同じ処理を行うが、 i 番目の子機の認証チップ1300__ i の場合とは、関数 $F_{s,c}$ を作用させる回数が t_0 であり、 t_i とは異なる。
- [0085] CPU1100は、S1306で親機の認証チップ1200からレスポンス R の値をレスポンス R_p として受信する。次に、S1307で t_i と t_0 を比較し、 t_i が t_0 より大きければS1308に進み、そうでなければS1309に進む。S1308では、 R_p に関数 $F_{s,c}$ を、 $t_i - t_0$ 回作用させた値と R_q が等しいかどうかを判定する。もし等しければ認証チップ1300__ i を含む i 番目の子機を正規品と判定する。そうでなければ、 i 番目の子機を正規品ではないと判定する。S1309では、 R_q に関数 $F_{s,c}$ を、 $t_0 - t_i$ 回作用させた値と R_p が等しいかどうかを判定してCPU100の処理を終了する。もし等しければ認証チップ1300__ i を含む i 番目の子機を正規品と判定してCPU100の処理を終了する。そうでなければ、 i 番目の子機を正規品ではないと判定してCPU100の処理を終了する。
- [0086] 前述のように、上記第2実施形態に従うシステム10では、関数 F のおよその内容を攻撃者が推定しながら、シリアルバス上のデータ盗聴を用いることで、変換手段0～変換手段 n の内容を完全に特定する攻撃に弱い可能性があった。関数 F は、ハードウェア実装において効率的な処理である必要があるため、LFSR、加算、XOR、剰余演算、ハッシュ関数などその手段は限られる。よって、攻撃者は、 F はこれらのいずれであるか、組み合わせであるかを推定しながら、その推定が実際に正しいかを、シリアルバス上のデータ盗聴を用いて判定することができる。例えば認証チップ0、認証チップ1の2つを用いて、かつ $t_0 = 0$ 、 $t_1 = 1$ にセットした場合、 $R_p = R$ 、 $R_q = F(R)$ となる。この R_p 、 R_q は、攻撃者にも容易に観察可能な値である。このとき、攻撃者は F の演算内容の推定を行い、その推定が実際に正しいかを、 $R_q = F(R_p)$ が成立するかどうか盗聴された R_p 、 R_q から判定することで確認することが出来てしまうおそれがあった。しかし、本実施形態に従うシステム20では、変換手段がチャレンジ C と、外部から観察できない

定数 S の影響を受けるため、変換のバリエーションを大きく増やすことができる。これにより、前述の通信データ盗聴を用いた変換手段の解読が攻撃者にとって困難となる。例えば、定数 S を128ビットパラメータとすることで、攻撃者の総当たり総数は 2^{128} となり、現実時間内では解読不可能となる。すなわち、本実施形態に従うシステム20を用いることで、第2実施形態に従うシステム10に対する、通信データ盗聴による変換手段の解読を用いた攻撃に対する耐性をさらに高めることができる。

[0087] (第4実施形態)

図16～20を参照して本発明の第4の実施形態に従う認証システム30を説明する。図16は本実施形態に従うシステム30を示す図、図17は本実施形態に従うシステム30で、認証チップ2200（認証チップ0）をCPUに装着する際にCPUで実行される処理のフローチャートを示す図、図18は本実施形態に従うシステム30で、認証チップ2200（認証チップ0）をCPU2100に装着する際に認証チップ2200で実行される処理のフローチャートを示す図、図19は、本実施形態に従うシステム30で、認証チップ2300__i（認証チップi）を含むi番目の子機を認証する際に、CPU2100で実行される処理のフローチャートを示す図、図20は、本実施形態に従うシステム30で、認証チップ2300__i（認証チップi）を含むi番目の子機を認証する際に、認証チップiで実行される処理のフローチャートを示す図である。

[0088] 第3実施形態に従うシステム20では、CPUに対する高度なリバースエンジニアリングを行った場合に、前述の共有値 S が特定できてしまう可能性がある。攻撃者は、CPUが実行するプログラムのROMコードを解析することで、変換手段 $F_{s,c}()$ を特定できる可能性がある。しかし、本実施形態では、CPUに対する高度なリバースエンジニアリングを行った場合でも、変換手段 $F_{s,c}()$ を特定することが難しいシステムが提供される。

[0089] 図16に示されているシステム30は、基本的には、図13に示されている前実施形態のシステム20と同じであるが、親機に装着されている認証チ

チップ2200（認証チップ0）とCPU2100の間のみで、固有値Iが共有されている。この固有値Iは、親機全体の製造時に、親機のCPU2100に認証チップ2200（認証チップ0）を最初に装着した時点でCPU2100と認証チップ2200の間で共有され、それぞれの内部の不揮発性メモリ2160、2232に格納され、以降書き換えられない値である。この値は、CPU固体間で異なる値である。子機は、CPUと認証チップ0が固有値の共有を完了した後に、親機に対して取り付けられるため、認証チップ2300__1～2300__n（認証チップ1～認証チップn）は、固有値Iを知ることはできない。

[0090] 認証チップ内で提供される複数の変換手段のうち、親機が用いる変換手段0だけは、S、Cを用いた変換手段 $F_{s,c}^{t_0}()$ を実行した後、この出力に対しさらに事前固有値Iをパラメータに含めた変換関数 $G_I()$ を実行することで、レスポンス $R_p = G_I(F_{s,c}^{t_0}(X))$ を出力する。変換手段1～nについては、実施形態3と同じ方法を用いる。すなわち、変換手段iは、レスポンス $R_q = F_{s,c}^{t_i}(X)$ を出力する。ただし、実施形態2、3と異なり、本実施形態においては、全ての $i = 1, 2, \dots, n$ に対して $t_0 > t_i$ 、すなわち t_0 が最大の繰り返し回数として設定されなければならないという制約がある。この制約は、CPU2100が正規品の子機からのレスポンス R_q を識別するための条件である。 $G_I(X)$ は、固有値Iをパラメータとした1入力1出力関数ならば、どのような関数でも良いが、例えば以下のような関数を用いることで、回路規模を抑えた効率的な実装が可能となる。

(8) パラメータつきXOR

(9) パラメータつき剰余演算

(10) パラメータつきハッシュ演算

(11) パラメータつきLFSR減算

[0091] 上記(8)の例としては、

[数4]

$$G_I(X) = X \oplus I$$

ただし I はパラメータ（固有値）、が含まれる。

[0092] 上記（９）の例としては、 $||$ はビット結合、 $q(X)$ は 128 ビット既約多項式、 mod を剰余として、

$$G_I(X) = (I || X) \text{ mod } q(X)$$

が含まれる。

上記（１０）の例としては、

[数5]

$$G_I(X) = \text{SHA1}(I \oplus X)$$

の最下位 128 ビットであり、ただし、 $\text{SHA1}(X)$ は X の $\text{SHA}-1$ ハッシュ関数出力、が含まれる。

[0093] 上記（１１）の例としては、

$$G_I(X) = \text{LFSR}(I, 75) - \text{LFSR}(X, 33)$$

が含まれる。

[0094] 本実施形態のシステム 30 の CPU 2100 は、共有値 S を記憶するメモリ 1140 および固有値 I を記憶するメモリ 2160 を含んでいる。

親機の認証チップ 2200 は、実施形態 3 の親機の認証チップ 1200 と類似の構成を有しているが、固有値 I を記憶する不揮発性メモリ 2232 を含むこと、変換 0 を行う回路 222a とセレクタ 223 の間に関数 G_I を用いて変換 G を行う回路 2230 を含む点が異なっている。変換 G を行う回路 2230 は、共有値 S を記憶するメモリ 2280 にも接続されている。

[0095] 子機の認証チップ 2300__1、...、2300__n は、固有値 I を記憶するメモリ 2232 を含まないことを除けば、親機の認証チップ 2200 と同一の構成を有している。

[0096] 親機側の変換手段 0 による変換では、親機製造時に CPU と認証チップ 0 の間で共有され、以後変更されない固有値 I に基づいて行われる。この値は、子機側から観察できないため、この値を攻撃者が推定することは困難である。例えば、固有値 I を 128 ビットパラメータとすることで、攻撃者の総当たり総数は 2^{128} となり、現実時間内では解読不可能となる。ただし、固有値 I は

CPU内部にも格納されている値であるため、CPUに対して高度なリバースエンジニアリングを行うことで、攻撃者が解読できる可能性がある。しかし、固有値IはCPU固体ごとに異なる値であるため、例えば高度な技法を用いて固有値Iを解読することに攻撃者が成功したとしても、認証チップ偽造上の大きな脅威にはならない。なぜなら、解読された固有値Iは、親機の固体ごとに異なる値であるため、攻撃者は特定固体の親機に対してのみ利用可能な偽造チップを製造することができるが、全ての固体に共通する偽造チップにはならない。本実施形態に従うシステム40では、CPUのROMコードの解析を用いた高度なリバースエンジニアリング技術に成功したとしても、親機に搭載される認証チップ0の変換手段はCPU固体ごとに異なるため、全ての親機に用いることが可能な子機認証チップの偽造品の量産は不可能となり、攻撃者は偽造品を市場に流通させることを防ぐことが出来る。

[0097] 図17および18を参照して、本実施形態に従うシステム30で、認証チップ0をCPUに装着する際の処理を説明する。

[0098] CPU2100は、S1501で、CPU2100内で発生させた乱数、現在時刻、CPUID、シリアル番号などを用いて、個体ごとにユニークな固有値Iを生成する。次にS1502で、固有値Iを親機の認証チップ（認証チップ0）2200に送信する。

[0099] S1551で、親機の認証チップ2200は、バス400を介してCPU2100から固有値Iを受信する。次に、S1552で、固有値Iを認証チップ2200の不揮発性メモリ2232に書き込み、S1553に進む。S1553では、親機に応答を送信する。この応答は、たとえば、不揮発性メモリ2232に書き込まれた値と固有値Iを比較した結果など、S1552の処理を確認できる情報である。

[0100] CPU2100は、この親機の認証チップ2200からの応答をS1503で受信する。次のS1504でCPU2100は、S1503で受信した親機の認証チップ2200からの応答が、親機の認証チップ2200での固有値Iの書き込み動作が正常に行われたことを示している場合にはS150

5に進み、固有値 I を CPU 2100 の不揮発性メモリ 2160 に書き込んで CPU 2100 の処理を終了する。S 1504 での判定が、S 1503 で受信した親機の認証チップ 2200 からの応答が、親機の認証チップ 2200 での固有値 I の書き込み動作が正常に行われなかったことを示している場合には、CPU 2100 はエラー終了する。

[0101] 図 19 は本実施形態に従うシステム 30 で、認証チップ i を含む i 番目の子機を認証する際に、CPU 2100 で実行される処理のフローチャートを示す図であり、図 20 は、本実施形態に従うシステムで、認証チップ i を含む i 番目の子機を認証する際に、認証チップ 2300__i で実行される処理のフローチャートを示す図である。

[0102] CPU 2100 は、S 1601 で認証対象の子機番号 i を決定し、S 1602 に進む。S 1602 では、チャレンジ C として用いるために乱数を生成する。S 1602 の次の S 1603 では、S 1602 で生成されたチャレンジ C を S 1601 で決定された i 番目の子機の認証チップ 2300__i に送信する。

[0103] i 番目の子機の認証チップ 2300__i では、S 1651 で親機の CPU 2100 によって送信されたチャレンジ C を受信する。次に、S 1652 では、認証チップ 300__i のレスポンス生成回路 324 中の秘密鍵記憶部 326 に記憶されている秘密鍵 K とチャレンジ C を用いて、レスポンスとしての出力値 $R = \text{Enc}(C, K)$ を生成する。次の S 1653 では、内部不揮発性メモリ 340 に記憶されたアドレス情報を元に、変換の繰り返し回数 t_i を決定する。S 1653 に引き続く S 1654 から S 1657 では、レスポンス R に t_i 回関数 $F_{s,c}$ を作用させる。関数 $F_{s,c}$ は、共有値 S およびチャレンジ C によって決まる。より詳細には、S 1654 で整数であるダミー変数 j に 0 を代入する。S 1655 でダミー変数の値を 1 つ増加させ、S 1656 で j と t_i を比較する。S 1656 で j が t_i より小さければ現在のレスポンス値に関数 $F_{s,c}$ を作用させ、レスポンス値を更新して、S 1655 に戻る。S 1656 で j が t_i より大きければ現在のレスポンス R の値を CPU 2

100に送信して、 i 番目の子機の認証チップ2300__ i の処理を終了する。

[0104] CPU2100は、S1604で i 番目の子機の認証チップ2300__ i からのレスポンス R の値をレスポンス R_0 として受信する。次に、S1605でチャレンジ C を今度は、親機の認証チップ2200に送信する。

[0105] 親機の認証チップ2200では、先の i 番目の子機の認証チップ2300__ i と同じ処理を行うが、 i 番目の子機の認証チップ2300__ i の場合とは、関数 $F_{s,c}$ を作用させる回数が t_0 であり、 t_i とは異なる。

[0106] CPU2100は、S1606で親機の認証チップ2200からのレスポンス R の値をレスポンス R_p として受信する。次に、S1607で R_p が $G_1(F_{s,c}^{t_0-t_i}(X))$ に等しいかどうかを判定する。もし等しければ i 番目の子機を正規品と判定してCPU100の処理を終了する。そうでなければ、 i 番目の子機を正規品ではないと判定してCPU2100の処理を終了する。

[0107] 上の説明では、S1603およびS1604の処理後にS1605およびS1606の処理を実行した。しかしながら、S1605およびS1606の処理の後でS1603およびS1604の処理を行っても良い。つまり、親機の認証チップ2200へのチャレンジ C の送信およびレスポンス R_p の受信と子機の認証チップ2300__ i へのチャレンジ C の送信およびレスポンス R_0 の受信の順序は任意である。

このような構成を用いることによって、高度なリバースエンジニアリングを用いた攻撃すらも回避した安全な認証を実現することができる。また、通信データの盗聴から変換手段を特定することを困難とすることができる。

[0108] (第5実施形態)

図21から23を参照して、本発明の第5の実施形態に従うシステム40を説明する。本実施形態は、図10に示されている第2実施形態のシステム10の特別な場合である。即ち、本実施形態では、変換関数は $F^0(X) = X$ と $F^1(X) = \text{SHA-1}$ 出力の最下位128ビットの2種である。

親機に認証チップ3200、子機に認証チップ3300が接続されている。チャレンジ、レスポンス、秘密鍵全て128ビットである。関数Fとして、SHA-1出力の最下位128ビットを利用する。関数Fの繰り返し回数は、 $t_0=1$ 、 $t_1=0$ である。すなわち、CPU100から送信されるチャレンジCに対して、AES暗号化を用いて出力値 $R=Enc(C, K)$ を生成し、認証チップのアドレスが親機ならば $R_p=F(R)$ を、子機ならば $R_q=R$ をレスポンスとしてCPU100に返信する。 R_p 、 R_q を受け取ったCPU100は、 $F(R_q)$ を計算し、 R_p と一致するならば子機を正規品と判定し、不一致ならば非正規品と判定する。

[0109] 図22は、本実施形態に従うシステム40のCPU100で実行される処理のフローチャートを示す図であり、図23は本実施形態に従うシステム40の親機の認証チップ3200および子機の認証チップ3300で実行される処理のフローチャートを示す図である。

[0110] CPU100は、S1801でチャレンジCとして用いるために乱数を生成し、S1802に進む。S1802では、S1802で生成されたチャレンジCを子機の認証チップ3300に送信する。

[0111] 子機の認証チップ3300では、S1851で親機のCPU100によって送信されたチャレンジCを受信する。次に、S1852では、認証チップ3300のレスポンス生成回路324中の秘密鍵記憶部326に記憶されている秘密鍵KとチャレンジCを用いて、レスポンスとしての出力値 $R=Enc(C, K)$ を生成する。次のS1853では、内部不揮発性メモリ340に記憶されたアドレス情報を元に、変換の繰り返し回数 t_1 を決定する。ここで $t_1=1$ である。S1853に引き続くS1854からS1857では、出力値Rに t_1 回関数 $F=SHA-1$ 出力の最下位128ビット、を作用させる。より詳細には、S1854で整数であるダミー変数jに0を代入する。S1855でダミー変数の値を1つ増加させ、S1856でjと t_1 を比較する。S1856でjが t_1 より小さければ現在のレスポンス値にS1857で関数Fを作用させ、レスポンス値を更新して、S1855に戻る。S1856

で j が t_i より大きければ現在のレスポンス R の値を CPU 100 に送信して、子機の認証チップ 3300 の処理を終了する。

[0112] CPU 100 は、S 1803 で子機の認証チップ 3300 からのレスポンス R の値をレスポンス R_o として受信する。次に、S 18054 チャレンジ C を今度は、親機の認証チップ 200 に送信する。

[0113] 親機の認証チップ 2200 では、先の子機の認証チップ 3300 と同じ処理を行うが、子機の認証チップ 3300 の場合とは、関数 F を作用させる回数が $t_o = 1$ であり、 $t_i = 0$ とは異なっている。

[0114] CPU 100 は、S 1805 で親機の認証チップ 2200 からのレスポンス R の値をレスポンス R_p として受信する。次に、S 1806 で、 R_p に関数 F を作用させた値 $F(R_p)$ と R_o が等しいかどうかを判定する。もし等しければ子機を正規品と判定して CPU 100 の処理を終了する。そうでなければ、子機を正規品ではないと判定する。

[0115] 上の説明では、CPU 2100 は、S 1802 および S 1803 の処理後に S 1804 および S 1805 の処理を実行した。しかしながら、S 1804 および S 1805 の処理の後で S 1802 および S 1803 の処理を行っても良い。

[0116] (第 6 実施形態)

図 24 から 26 を参照して、本発明の第 6 の実施形態に従うシステム 50 を説明する。本実施形態は、図 13 に示されている第 3 実施形態のシステム 20 の特別な場合である。即ち、本実施形態では、子機の認証チップは子機 1 の認証チップ 4300__1 と子機 2 の認証チップ 4300__2 の 2 つを含み、レスポンス生成回路 324 で生成されたレスポンスに作用する関数として、

[数 6]

$$F^0(x) = x \text{ と } F^1(x) = (S + X) \oplus C, F^2(x) = F^1(F^1(x))$$

が用意されている。

[0117] 本実施形態の構成を図 24 に示す。親機に認証チップ 4200、子機に 2

個の認証チップ4300__1、4300__2が接続されている。チャレンジ、レスポンス、秘密鍵全て128ビットである。関数 $F_{s,c}$ として、パラメータ付XOR加算

[数7]

$$F_{s,c}(X) = (S + X) \oplus C$$

を利用する。関数 $F_{s,c}$ の繰り返し回数は、 $t_0=0$ 、 $t_1=1$ 、 $t_2=2$ である。すなわち、CPU1100から送信されるチャレンジCに対して、レスポンス生成回路224、324においてAES暗号化を用いて $R = \text{Enc}(C, K)$ を生成し、認証チップのアドレスが親機ならば $R_p = R$ を、子機1ならば $R_q = F_{s,c}(R)$ を、子機2なら $R_q = F_{s,c^2}(R)$ を、レスポンスとしてCPU100に返信する。 R_p 、 R_q を受け取ったCPU1100は、 $F_{s,c}^{t_i-t_0}(R_p)$ を計算し、 R_q と一致するならば子機を正規品と判定し、不一致ならば非正規品と判定する。すなわち、 $i=1$ の場合は $F_{s,c}(R_p) = R_q$ であるか、 $i=2$ の場合は $F_{s,c^2}(R_p) = R_q$ であるかを判定する。

[0118] 本実施形態の認証システム50の処理を説明する。

図25は、本実施形態に従うシステム50のCPU1100で実行される処理のフローチャートを示す図であり、図26は本実施形態に従うシステム50の親機または子機の認証チップ4200、4300__i ($i=1, 2$)で実行される処理のフローチャートを示す図である。

[0119] CPU100は、S2001で認証対象の子機番号iを決定し、S2002に進む。ここで $i=1$ または2である。S2002では、チャレンジCとして用いるために乱数を生成する。S2002の次のS2003では、S2002で生成されたチャレンジCをS2001で決定されたi番目の子機の認証チップ4300__iに送信する。

[0120] i番目の子機の認証チップ4300__iでは、S2051で親機のCPU1100によって送信されたチャレンジCを受信する。次に、S2052では、認証チップ4300__iのレスポンス生成回路324中の秘密鍵記憶部326に記憶されている秘密鍵KとチャレンジCを用いて、レスポンスとし

での出力値 $R = \text{Enc}(C, K)$ を生成する。次の S2053 では、内部不揮発性メモリ 340 に記憶されたアドレス情報を元に、変換の繰り返し回数 t_i を決定する。S2053 に引き続く S2054 から S2057 では、レスポンス R に t_i 回関数 $F_{s,c}$ を作用させる。関数 $F_{s,c}$ は、共有値 S およびチャレンジ C によって決まる。より詳細には、S2054 で整数であるダミー変数 j に 0 を代入する。S2055 でダミー変数の値を 1 つ増加させ、S2056 で j と t_i を比較する。S2056 で j が t_i より小さければ現在のレスポンス値に S2057 で関数 $F_{s,c}$ を作用させ、レスポンス値を更新して、S2055 に戻る。S2056 で j が t_i より大きければ現在のレスポンス R の値を CPU 1100 に送信して、 i 番目の子機の認証チップ 1300__ i の処理を終了する。

[0121] CPU 1100 は、S2004 で i 番目の子機の認証チップ 4300__ i からレスポンス R の値をレスポンス R_o として受信する。次に、S2005 でチャレンジ C を今度は、親機の認証チップ 4200 に送信する。

[0122] 親機の認証チップ 4200 では、先の i 番目の子機の認証チップ 4300__ i と同じ処理を行うが、 i 番目の子機の認証チップ 4300__ i の場合、関数 $F_{s,c}$ を作用させる回数が t_o であり、 t_i とは異なる。今の場合、 $t_o = 0$ であるので、関数 $F_{s,c}$ は作用させない。

[0123] CPU 1100 は、S2006 で親機の認証チップ 4200 からのレスポンス R の値をレスポンス R_p として受信する。次に S2007 では、 R_o が $F_{s,c}^{t_o - t_i}(R_p)$ に等しいかどうかを判定する。もし等しければ i 番目の子機を正規品と判定して CPU 1100 の処理を終了する。そうでなければ、 i 番目の子機を正規品ではないと判定して CPU 1100 の処理を終了する。

[0124] (第 7 実施形態)

図 27 から 31 を参照して、本発明の第 7 の実施形態に従うシステム 60 を説明する。本実施形態は、図 16 に示されている第 4 実施形態のシステム 30 の特別な場合である。即ち、本実施形態では、子機の認証チップは子機 1 の認証チップ 5300__1 と子機 2 の認証チップ 5300__2 の 2 つを合

み、レスポンス生成回路 3 2 4 で生成されたレスポンスに作用する変換関数 $F_{s,c}$ としてパラメータ付剰余算 $F_{s,c}(X) = (S || C || X) \bmod q(X)$ 、変換関数 G_I として、

[数8]

$$G_I(X) = \text{SHA1}(I \oplus X)$$

の最下位 1 2 8 ビットを用いる。

[0125] 親機に認証チップ 5 2 0 0、子機に 2 個の認証チップ 5 3 0 0__1、5 3 0 0__2 が接続されている。チャレンジ、レスポンス、秘密鍵全て 1 2 8 ビットである。関数 $F_{s,c}$ としてパラメータ付剰余算 $F_{s,c}(X) = (S || C || X) \bmod q(X)$ を、関数 G_I としてパラメータつきハッシュ演算：

[数9]

$$\text{SHA1}(I \oplus X)$$

の最下位 1 2 8 ビット、を利用する。ただし、 $q(X)$ は 1 2 8 ビット既約多項式である。関数 $F_{s,c}$ の繰り返し回数は、 $t_0=2$ 、 $t_1=1$ 、 $t_2=0$ である。つまり、CPU 2 1 0 0 から送信されるチャレンジ C に対して、AES 暗号化を用いて $R = \text{Enc}(C, K)$ を生成し、認証チップのアドレスが親機ならば $R_P = G_I(F_{s,c}^2(R))$ を、子機 1 ならば $R_Q = F_{s,c}(R)$ を、子機 2 なら $R_Q = R$ を、レスポンスとして CPU 2 1 0 0 に返信する。 R_P 、 R_Q を受け取った CPU 2 1 0 0 は、 $G_I(F_{s,c}^{2-t_i}(R_Q))$ を計算し、 R_P と一致するならば子機を正規品と判定し、不一致ならば非正規品と判定する。すなわち、 $i=1$ の場合は $G_I(F_{s,c}(R_Q)) = R_P$ であるか、 $i=2$ の場合は $G_I(F_{s,c}^2(R_Q)) = R_P$ であるかを判定する。

[0126] 図 2 8 および 2 9 を参照して、本実施形態に従うシステム 6 0 で、認証チップ 5 2 0 0 を CPU 2 1 0 0 に装着する際の処理を説明する。

CPU 2 1 0 0 は、S 2 2 0 1 で、CPU 2 1 0 0 内で発生させた乱数、現在時刻、CPU ID、シリアル番号などを用いて、CPU 固体ごとにユニークな固有値 I を生成する。次に S 2 2 0 2 で、固有値 I を親機の認証チップ（認証チップ 0）5 2 0 0 に送信する。

- [0127] S 2 2 5 1 で、親機の認証チップ 5 2 0 0 は、バス 4 0 0 を介して CPU 2 1 0 0 から固有値 I を受信する。次に、S 2 2 5 2 で、固有値 I を認証チップ 5 2 0 0 の不揮発性メモリ 2 2 3 2 に書き込み、S 2 2 5 3 に進む。S 2 2 5 3 では、親機に応答を送信する。この応答は、たとえば、不揮発性メモリ 2 2 3 2 に書き込まれた値と固有値 I を比較した結果など、S 2 2 5 2 の処理を確認できる情報である。
- [0128] CPU 2 1 0 0 は、この親機の認証チップ 5 2 0 0 からの応答を S 2 2 0 3 で受信する。次の S 2 2 0 4 で CPU 2 1 0 0 は、S 2 2 0 3 で受信した親機の認証チップ 5 2 0 0 からの応答が、親機の認証チップ 5 2 0 0 での固有値 I の書き込み動作が正常に行われたことを示している場合には S 2 2 0 5 に進み、固有値 I を CPU 2 1 0 0 の不揮発性メモリ 2 1 6 0 に書き込んで CPU 2 1 0 0 の処理を終了する。S 2 2 0 4 での判定が、S 2 2 0 3 で受信した親機の認証チップ 5 2 0 0 からの応答が、親機の認証チップ 5 2 0 0 での固有値 I の書き込み動作が正常に行われなかったことを示している場合には、CPU 2 1 0 0 はエラー終了する。
- [0129] 図 3 0 は本実施形態に従うシステム 6 0 で、認証チップ（認証チップ～2）5 2 0 0、5 3 0 0__1、5 3 0 0__2 を含む親機および i 番目の子機を認証する際に、CPU 2 1 0 0 で実行される処理のフローチャートを示す図であり、図 3 1 は、本実施形態に従うシステムで、認証チップ i を含む親機または i 番目の子機を認証する際に、認証チップ 5 2 0 0、5 3 0 0__i（i = 1、2）で実行される処理のフローチャートを示す図である。
- [0130] CPU 2 1 0 0 は、S 2 3 0 1 で認証対象の子機番号 i を決定し、S 2 3 0 2 に進む。S 2 3 0 2 では、チャレンジ C として用いるために乱数を生成する。S 2 3 0 2 の次の S 2 3 0 3 では、S 2 3 0 2 で生成されたチャレンジ C を S 2 3 0 1 で決定された i 番目の子機の認証チップ 5 3 0 0__i に送信する。
- [0131] i 番目の子機の認証チップ 5 3 0 0__i では、S 2 3 5 1 で親機の CPU 2 1 0 0 によって送信されたチャレンジ C を受信する。次に、S 2 3 5 2 で

は、認証チップ5300__iのレスポンス生成回路324中の秘密鍵記憶部326に記憶されている秘密鍵KとチャレンジCを用いて、レスポンスとしての出力値 $R = \text{Enc}(C, K)$ を生成する。次のS2353では、内部不揮発性メモリ340に記憶されたアドレス情報を元に、変換の繰り返し回数 t_i を決定する。S2353に引き続くS2354からS2357では、レスポンスRに t_i 回関数 $F_{s,c}$ を作用させる。関数 $F_{s,c}$ は、共有値SおよびチャレンジCによって決まる。より詳細には、S1654で整数であるダミー変数jに0を代入する。S2355でダミー変数の値を1つ増加させ、S2356でjと t_i を比較する。S2356でjが t_i より小さければS2357で現在のレスポンス値に関数 $F_{s,c}$ を作用させ、レスポンス値を更新して、S2355に戻る。S2356でjが t_i より小さくなければ、S2358に進む。S2358では、現在認証中のチップが親機であるかを判定する。今の場合、認証チップは親機のものではないので、現在のレスポンスRの値をCPU2100に送信して、i番目の子機の認証チップ5300__iの処理を終了する。

[0132] CPU2100は、S2304でi番目の子機の認証チップ2300__iからのレスポンスRの値をレスポンス R_Q として受信する。次に、S2305でチャレンジCを今度は、親機の認証チップ5200に送信する。

[0133] 親機の認証チップ5200では、先のi番目の子機の認証チップ5300__iと同じ処理を行うが、i番目の子機の認証チップ5300__iの場合とは、関数 $F_{s,c}$ を作用させる回数が t_0 であり、 t_i とは異なること、S2358での現在認証中のチップが親機であるかの判定で、今の場合、親機であるから、S2359に進むことが異なる。S2359では、固有値Iと G_I 計算回路2230を用いて、 $G_I(F_{s,c}^{2^{-t_i}}(R_Q))$ を計算する。

[0134] CPU2100は、S2306で親機の認証チップ2200からのレスポンスRの値をレスポンス R_P として受信する。次に、S1607で R_P が $G_I(F_{s,c}^{2^{-t_i}}(R_Q))$ に等しいかどうかを判定する。もし等しければi番目の子機を正規品と判定してCPU2100の処理を終了する。そうでなければ

、 i 番目の子機を正規品ではないと判定してCPU 2100の処理を終了する。

請求の範囲

[請求項1]

第1の認証チップを含む第1の装置を制御するプロセッサにより、
第2の認証チップを含む第2の装置を認証する認証方法であって、
乱数を生成し、

前記乱数を前記第2の認証チップおよび前記第2の認証チップに送信し、

前記第1の認証チップに秘密鍵として記憶された整数と、前記乱数に対して暗号化をするための暗号化関数を作用させて生成された出力値に、前記第1の認証チップに設定された値で決まる第1の変換関数を作用させて得られる第1のレスポンス値を前記第1の認証チップから受信し、

前記第2の認証チップに秘密鍵として記憶された前記第1の認証チップに秘密鍵として記憶された前記整数と同じ整数と、前記乱数に対して暗号化をするための暗号化関数を作用させて生成された前記出力値に、前記第2の認証チップに設定された値で決まる第2の変換関数を作用させて得られる第2のレスポンス値を前記第2の認証チップから受信し、

前記第1のレスポンス値に、前記第1の認証チップに設定された値と前記第2の認証チップに設定された値の大きさの差の数で決まる第3の変換関数を作用させて得られる値と前記第2のレスポンス値を比較して、または前記第2のレスポンス値に前記第3の変換関数を作用させて得られる値と前記第1のレスポンス値を比較して、前記第2の装置を認証する、
認証方法。

[請求項2]

前記第1の認証チップに設定された値で決まる前記第1の変換関数は所定の関数の冪乗であり、その指数は前記第1の認証チップに設定された値で決まる第1の指数であり、前記第2の認証チップに設定された値で決まる第2の変換関数は前記所定の関数の冪乗であり、その

指数は前記第 2 の認証チップに設定された値で決まり前記第 1 の指数と異なる第 2 の指数であり、前記第 1 の認証チップに設定された値と前記第 2 の認証チップに設定された値の大きさの差の数で決まる前記第 3 の変換関数は、前記所定の関数の冪乗であり、その指数は、前記第 1 の指数と前記第 2 の指数の差の絶対値に等しい第 3 の指数である、請求項 1 の方法。

[請求項3] 前記所定の関数はハッシュ関数である、請求項 2 の方法。

[請求項4] 前記所定の関数は定数ビットシフトである、請求項 2 の方法。

[請求項5] 前記所定の関数は左 1 ビットシフト後、 $GF(2^m)$ 上の特性多項式の剰余演算処理を定数ビット分の繰り返しである、請求項 2 の方法。

[請求項6] 前記第 1 の変換関数は前記第 1 の認証チップに設定された値と予め前記第 2 の認証チップと共通に設定された整数である共有値により決定され、前記第 2 の変換関数は前記第 2 の認証チップに設定された値と前記共有値により決定される、請求項 1 乃至 5 のいずれか一項の方法。

[請求項7] 前記第 1 のレスポンス値は、前記出力値に、前記第 1 の変換関数に加えて、前記プロセッサに特有の値である固有値で決まる第 4 の変換関数を作用させて得られる、請求項 1 乃至 6 のいずれか一項の方法。

[請求項8] 前記第 2 の認証チップに設定された値および前記共有値で決まる前記第 2 の変換関数は、パラメータ付 XOR 演算である、請求項 6 又は 7 の方法。

[請求項9] 前記第 2 の認証チップに設定された値および前記共有値で決まる前記第 2 の変換関数は、パラメータつき剰余演算である、請求項 6 又は 7 の方法。

[請求項10] 前記第 2 の認証チップに設定された値および前記共有値で決まる前記第 2 の変換関数は、パラメータハッシュ演算の最下位の定数ビット幅のデータ値を取る出す演算である、請求項 6 又は 7 の方法。

- [請求項11] 前記第2の認証チップに設定された値および前記共有値で決まる前記第2の変換関数は、パラメータ付定数ビットシフト、XORおよび減算の組み合わせの演算である、請求項6又は7の方法。
- [請求項12] 前記プロセッサ、前記第1の認証チップ、および前記第2の認証チップは互いの通信のためにシリアルバス接続されており、前記親機認証チップおよび前記子機認証チップは前記通信のためにそれぞれ第1のアドレスおよび第2のアドレスを有し、前記親機認証チップに設定された値および前記子機認証チップに設定された値は、それぞれ前記第1のアドレスおよび前記第2のアドレスから決まる、請求項1乃至12のいずれか一項の方法。
- [請求項13] 第1の装置が第2の装置の認証を行う認証システムであって、
前記第1の装置に含まれ、乱数を生成し、前記乱数を前記第1の装置が含む第1の認証チップと前記第2の装置が含む第2の認証チップとに送信し、前記第1の認証チップと前記第2の認証チップから、前記乱数に対する応答として第1のレスポンス値と第2のレスポンス値とをそれぞれ受信し、前記第1のレスポンス値と前記第2のレスポンス値を用いて前記第2の装置を認証する処理を実行するプロセッサと、
前記乱数を受信する第1の乱数受信手段と、秘密鍵としての整数を記憶する第1の記憶手段と、秘密鍵として記憶された整数および前記乱数に対して暗号化をするための暗号化関数を作用させて出力値を生成する第1の暗号演算手段と、生成された前記出力値に、第1の設定された値で決まる第1の変換関数を作用させて前記第1のレスポンス値を生成する第1のレスポンス値生成手段と、を含み、前記プロセッサに接続されている第1の認証チップと、
前記乱数を受信する第2の乱数受信手段と、秘密鍵としての整数を記憶する第2の記憶手段と、秘密鍵として記憶された整数および前記乱数に対して暗号化をするための暗号化関数を作用させて前記出力値

を生成する第2の暗号演算手段と、前記出力値に、第2の設定された値で決まる第2の変換関数を作用させて前記第2のレスポンス値を生成する第2のレスポンス値生成手段と、を含み、少なくとも前記プロセッサに接続されている第2の認証チップと、を含み、

前記プロセッサは、前記第1のレスポンス値に、前記第1の設定された値と前記第2の設定された値の大きさの差の絶対値で決まる第3の変換関数を作用させて得られる数と前記第2のレスポンス値を比較して、または前記第2のレスポンス値に前記第3の変換関数を作用させて得られる数と前記第1のレスポンス値を比較して、前記子機を認証する処理を実行すること、を特徴とする認証システム。

[請求項14] 前記プロセッサと前記第1の認証チップおよび前記第2の認証チップをシリアルバス接続で接続するバスを含む、請求項13の認証システム。

[請求項15] 前記所定の関数は、ハッシュ関数、定数ビットシフト、左1ビットシフト後、 $GF(2^m)$ 上の特性多項式の剰余演算処理を定数ビット分の繰り返しからなる群から選択される、請求項13または14の認証システム。

[請求項16] 前記第1の認証チップおよび前記第2の認証チップは前記通信のためにそれぞれ第1のアドレスおよび第2のアドレスを有し、前記第1の認証チップに設定された値および前記第2の認証チップに設定された値は、それぞれ前記第1のアドレスおよび前記第2のアドレスから決まる、請求項13乃至15のいずれか一項の認証システム。

[請求項17] 前記第1の認証チップに設定された値で決まる前記第1の変換関数は所定の関数の冪乗であり、その指数は前記第1の認証チップに設定された値で決まる第1の指数であり、前記第2の認証チップに設定された値で決まる第2の変換関数は前記所定の関数の冪乗であり、その

指数は前記第 2 の認証チップに設定された値で決まり前記第 1 の指数と異なる第 2 の指数である、請求項 13 乃至 16 のいずれか一項の認証システム。

[請求項18] 前記第 1 の変換関数は前記第 1 の認証チップに設定された値および予め前記第 2 の認証チップと共通に設定された整数である共有値で決まり、前記第 2 の変換関数は前記第 2 の認証チップに設定された値および前記共有値で決まる、請求項 13 乃至 17 のいずれか一項の認証システム。

[請求項19] 前記第 1 のレスポンス値は、前記出力値に、前記第 1 の変換関数に加えて、前記プロセッサに特有の値である固有値で決まる第 4 の変換関数を作用させて得られる、請求項 13 乃至 18 のいずれか一項の認証システム。

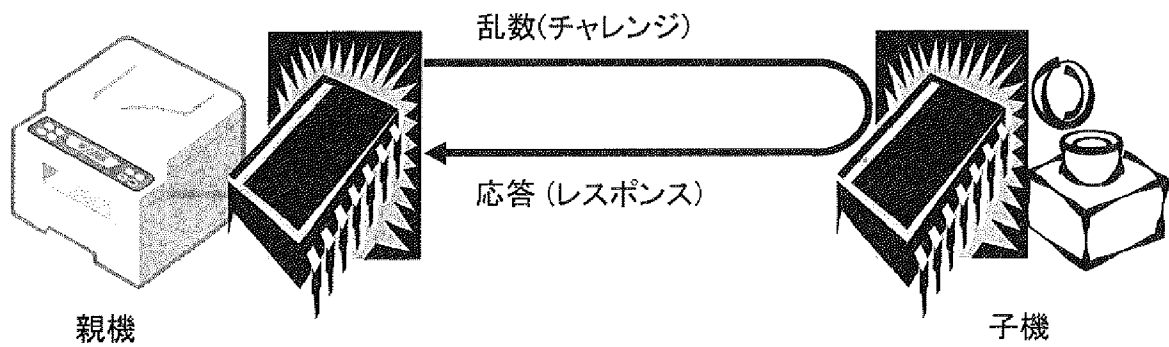
[請求項20] 前記第 2 の変換関数は、パラメータ付 XOR 演算、パラメータつき剰余演算、パラメータハッシュ演算の最下位の定数ビット幅のデータ値を取る出す演算、パラメータ付定数ビットシフト、XOR および減算の組み合わせの演算からなる群から選択される、請求項 13 乃至 20 のいずれか一項の認証システム。

[請求項21] 整数である秘密鍵を記憶するメモリと、
整数を受信する受信手段と、
前記整数および前記秘密鍵に暗号化のための暗号化関数を作用させて生成された出力値に、変換関数を作用させて得られるレスポンス値を生成するレスポンス値生成手段と、
前記レスポンス値を送信する送信手段、
を含む認証チップ。

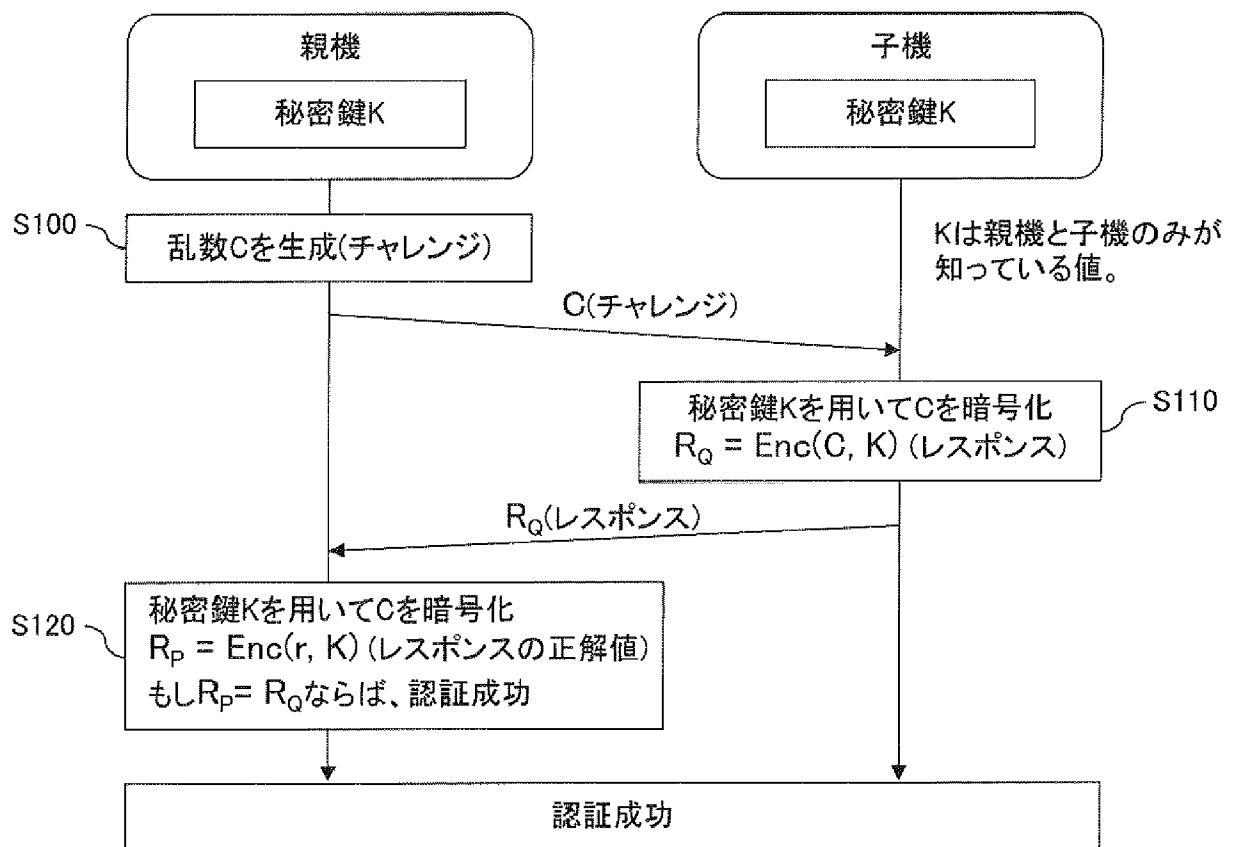
[請求項22] 前記レスポンス値生成手段は、さらに、前記通信におけるアドレスを記憶するアドレス記憶手段を含み、
前記変換関数は、前記レスポンス値は前記アドレスで決まる、
請求項 21 の認証チップ。

- [請求項23] 前記レスポンス値生成手段は、それぞれ関数を記憶するメモリを含む複数の演算部を含み、
前記変換関数は、前記複数の関数から選択される、
請求項22の認証チップ。
- [請求項24] 前記変換関数は、前記変換関数は、ハッシュ関数、定数ビットシフト、左1ビットシフト後、 $GF(2^m)$ 上の特性多項式の剰余演算処理を定数ビット分の繰り返しからなる群から選択される、
請求項21乃至23のいずれか一項の認証チップ。
- [請求項25] さらに、
整数である共有値を記憶するメモリを含み、
前記変換関数は前記アドレスおよび前記共有値で決まる、
請求項21乃至24のいずれか一項の認証チップ。
- [請求項26] 前記認証チップはプロセッサに接続され、
前記レスポンス値は、前記出力値に前記変換関数を作用させ、さらに前記プロセッサに特有の値である固有値で決まる別の変換関数を作用させて得られる、
請求項21乃至25のいずれか一項の認証チップ。
- [請求項27] 前記プロセッサに特有の値である固有値で決まる前記別の変換関数は、パラメータ付XOR演算、パラメータつき剰余演算、パラメータハッシュ演算の最下位の定数ビット幅のデータ値を取る出す演算、パラメータ付定数ビットシフト、XORおよび減算の組み合わせの演算からなる群から選択される、
請求項25の認証チップ。

[図1]

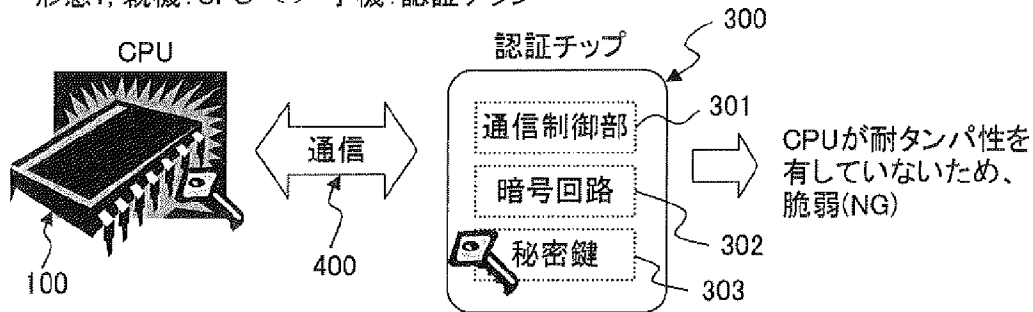


[図2]

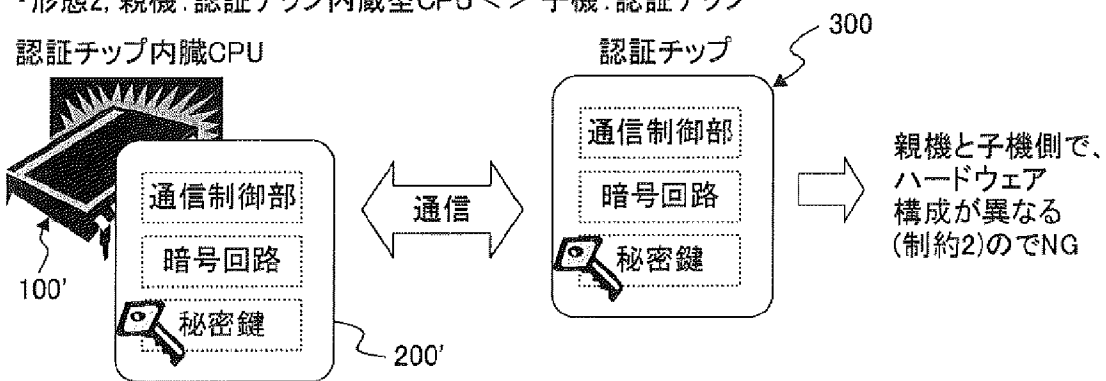


[図3]

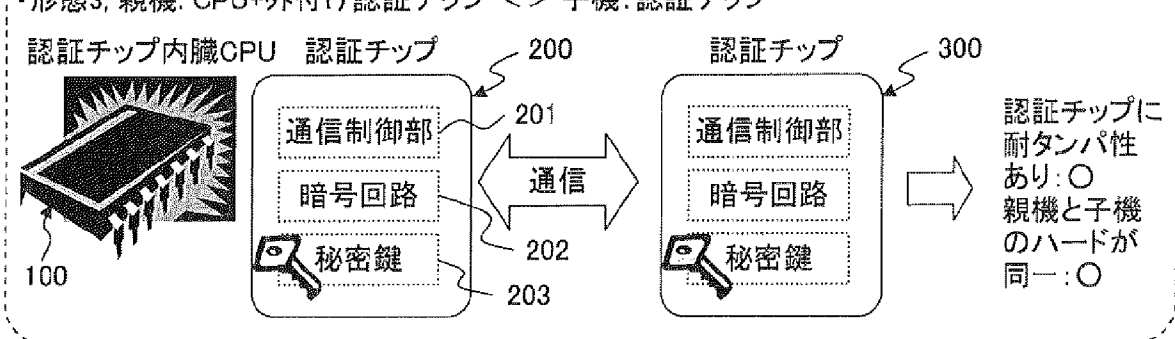
・形態1, 親機: CPU <=> 子機: 認証チップ



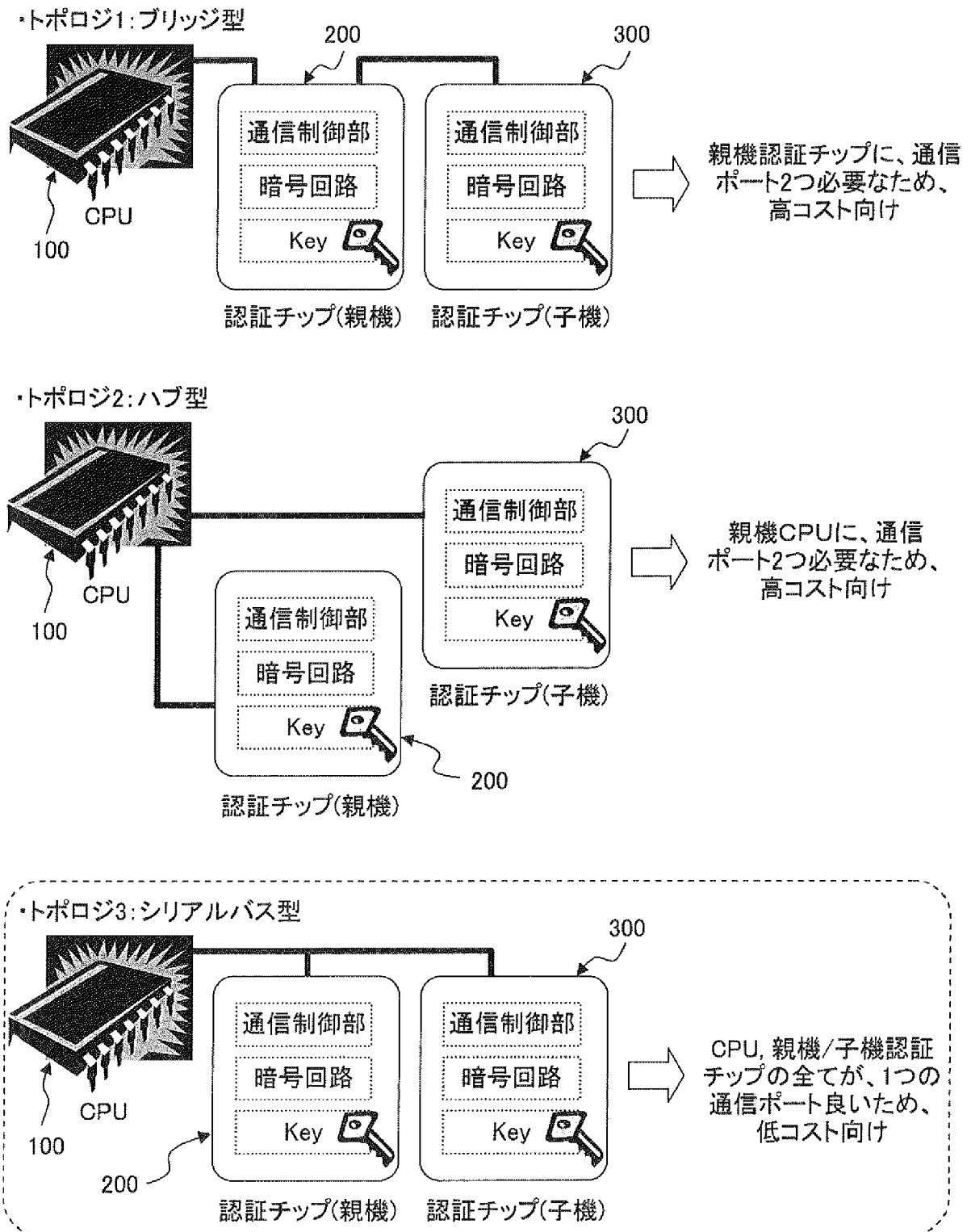
・形態2, 親機: 認証チップ内蔵型CPU <=> 子機: 認証チップ



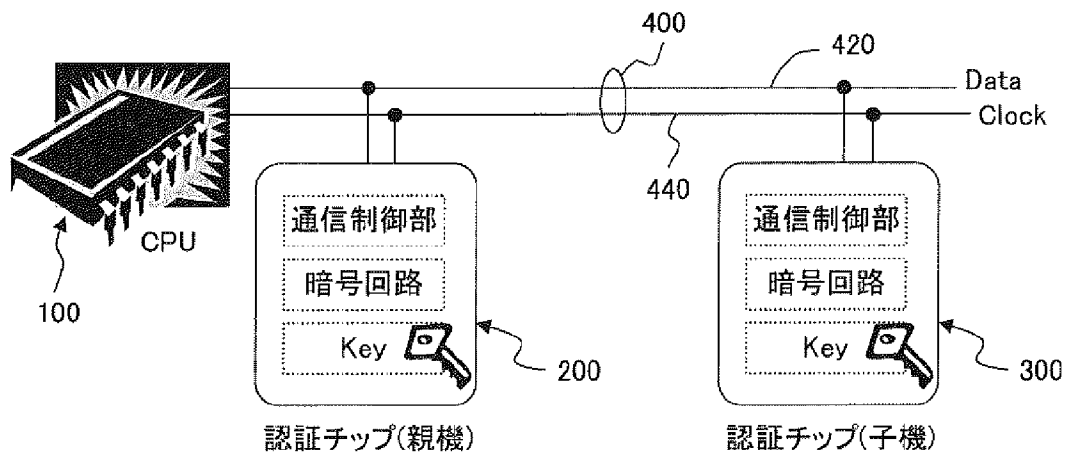
・形態3, 親機: CPU+外付け認証チップ <=> 子機: 認証チップ



[図4]

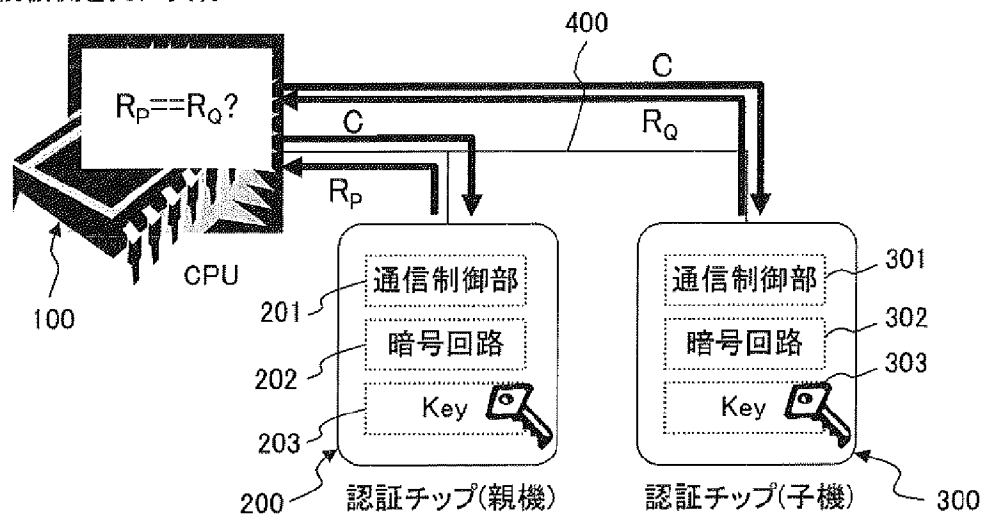


[図5]

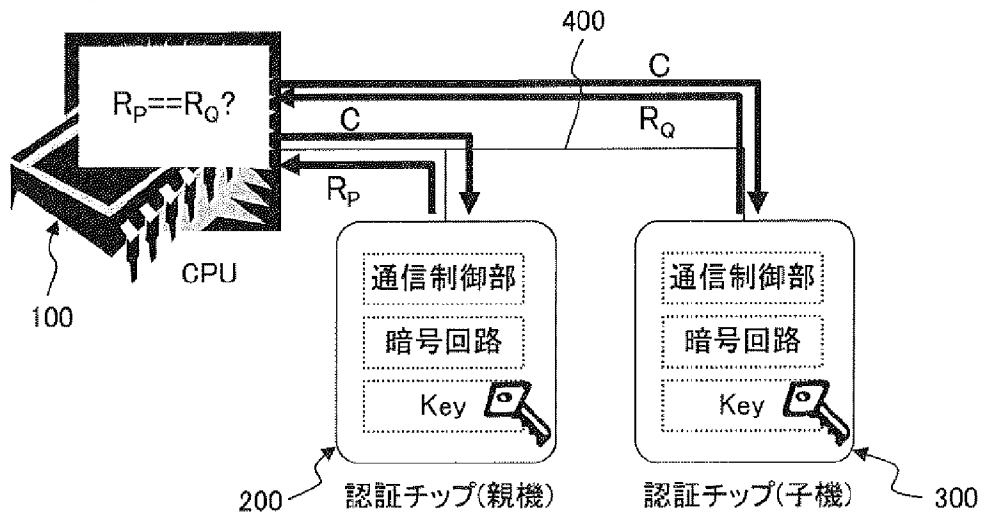


[図6]

親機側を先に実行

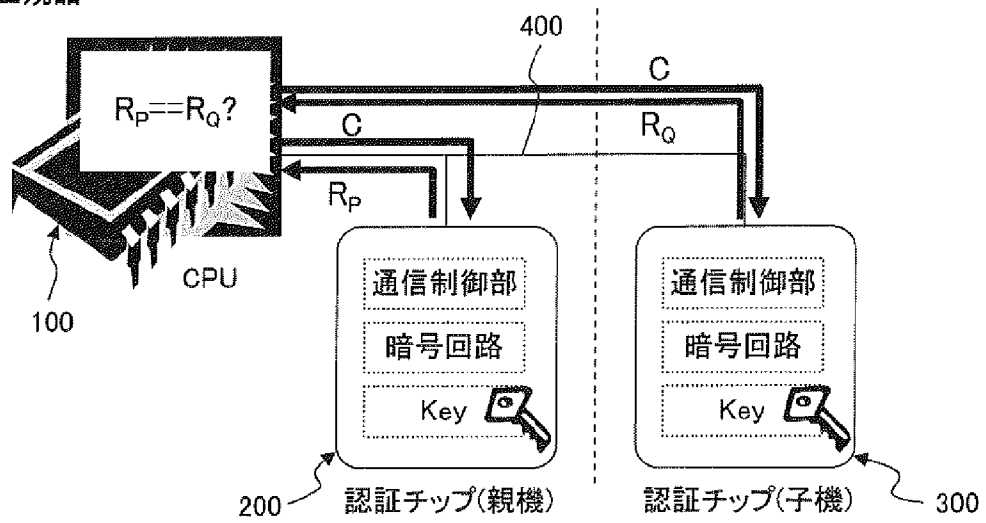


子機側を先に実行

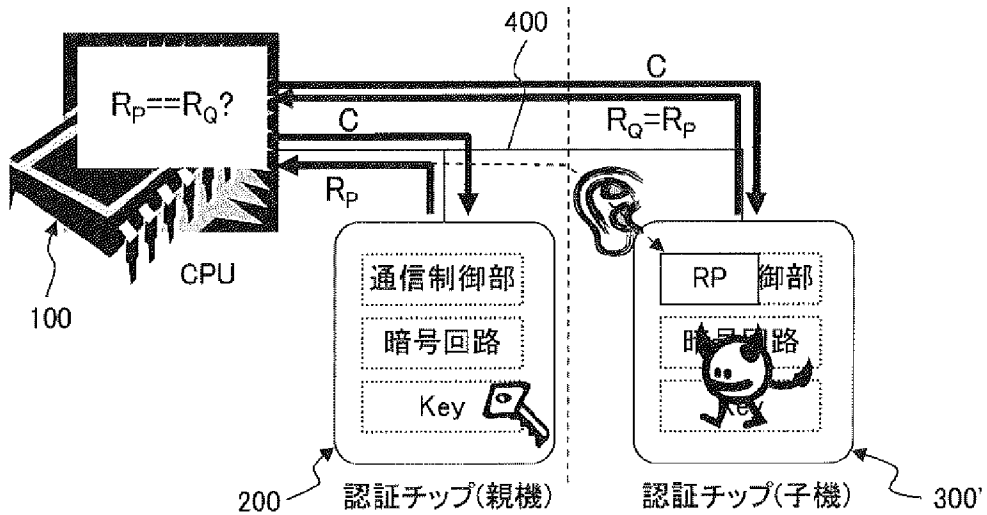


[図7]

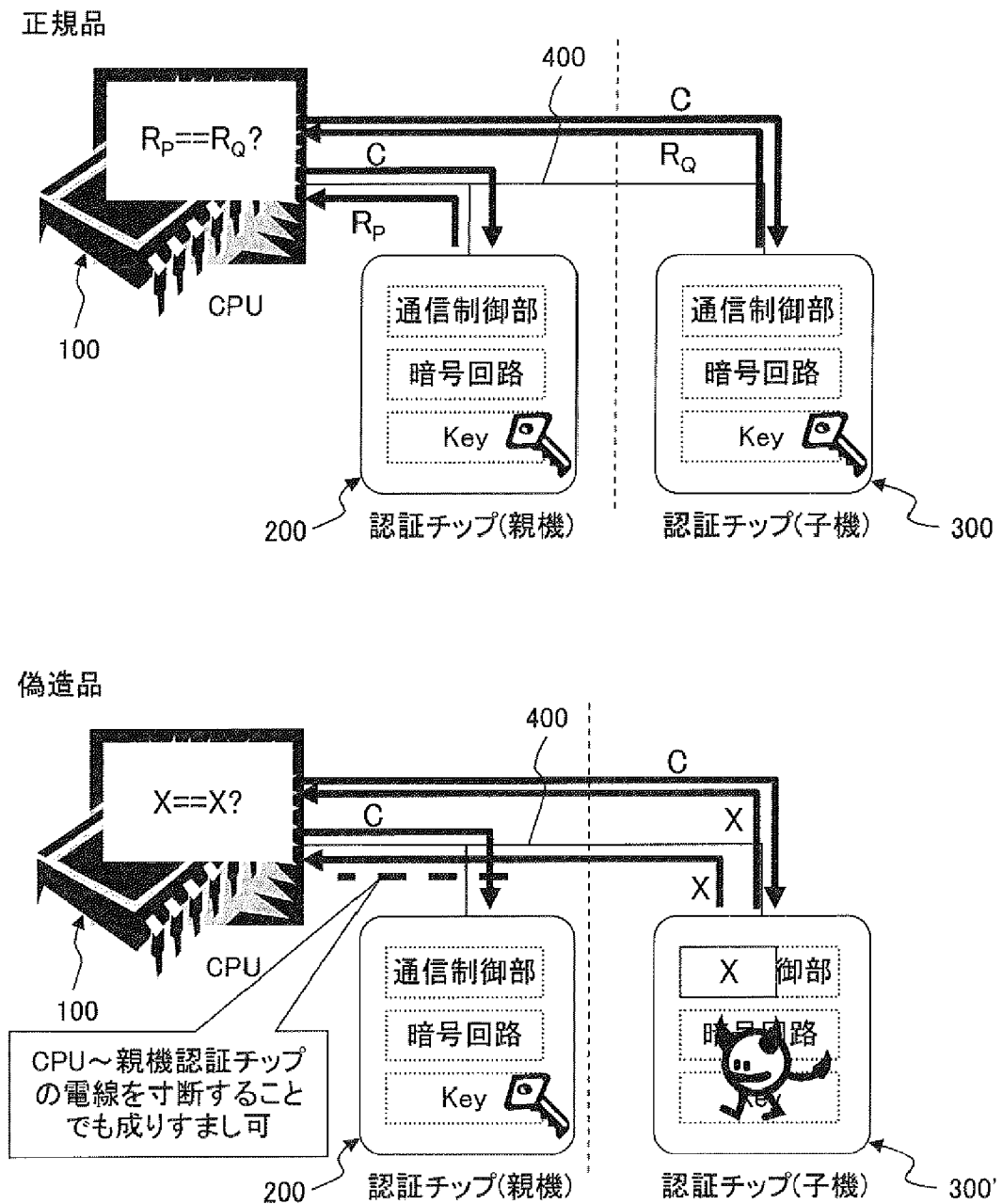
正規品



偽造品

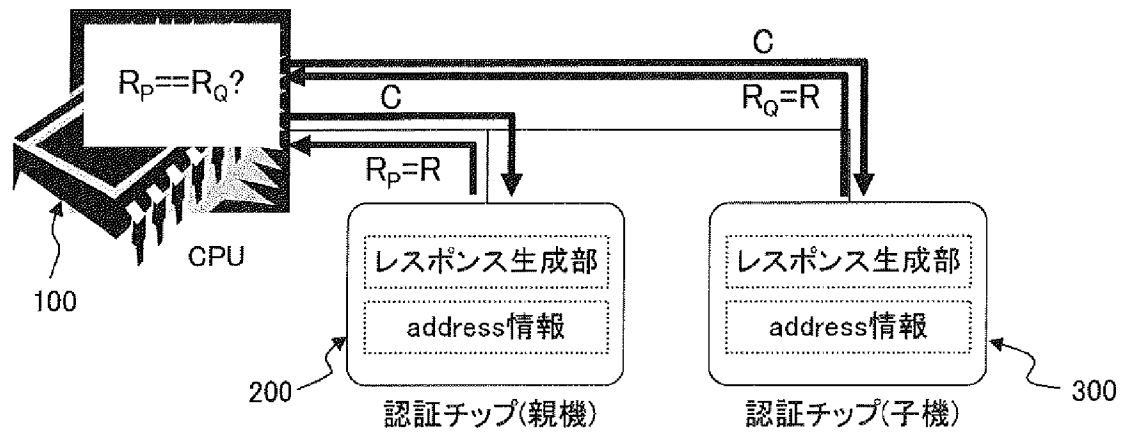


[図8]

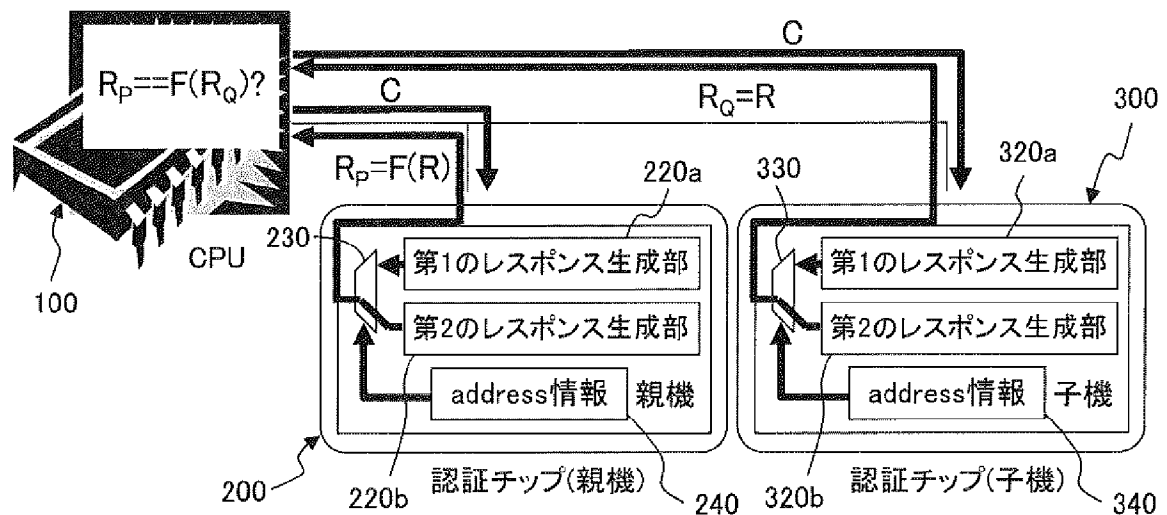


[図9]

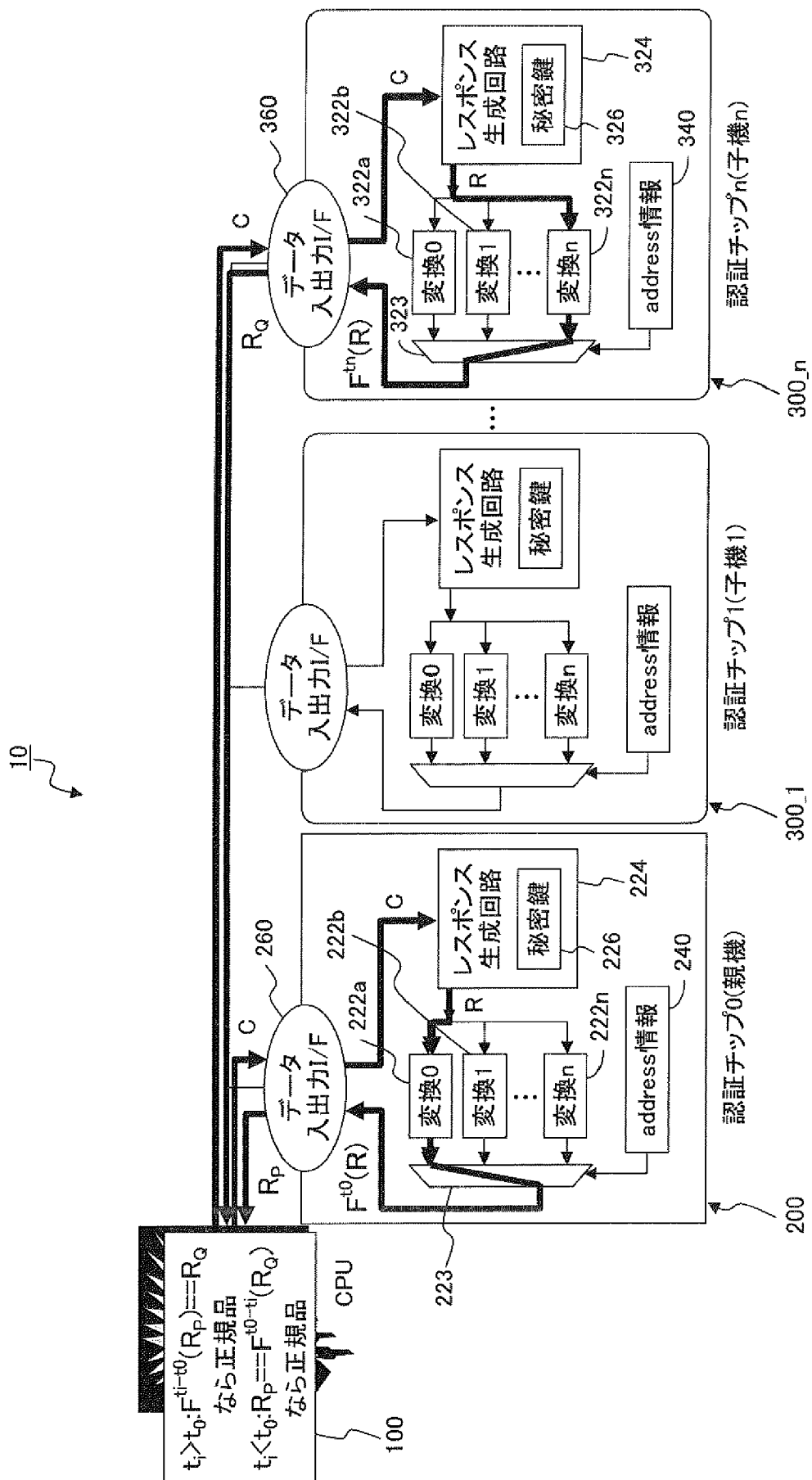
従来：親機、子機ともに同じ動作脆弱



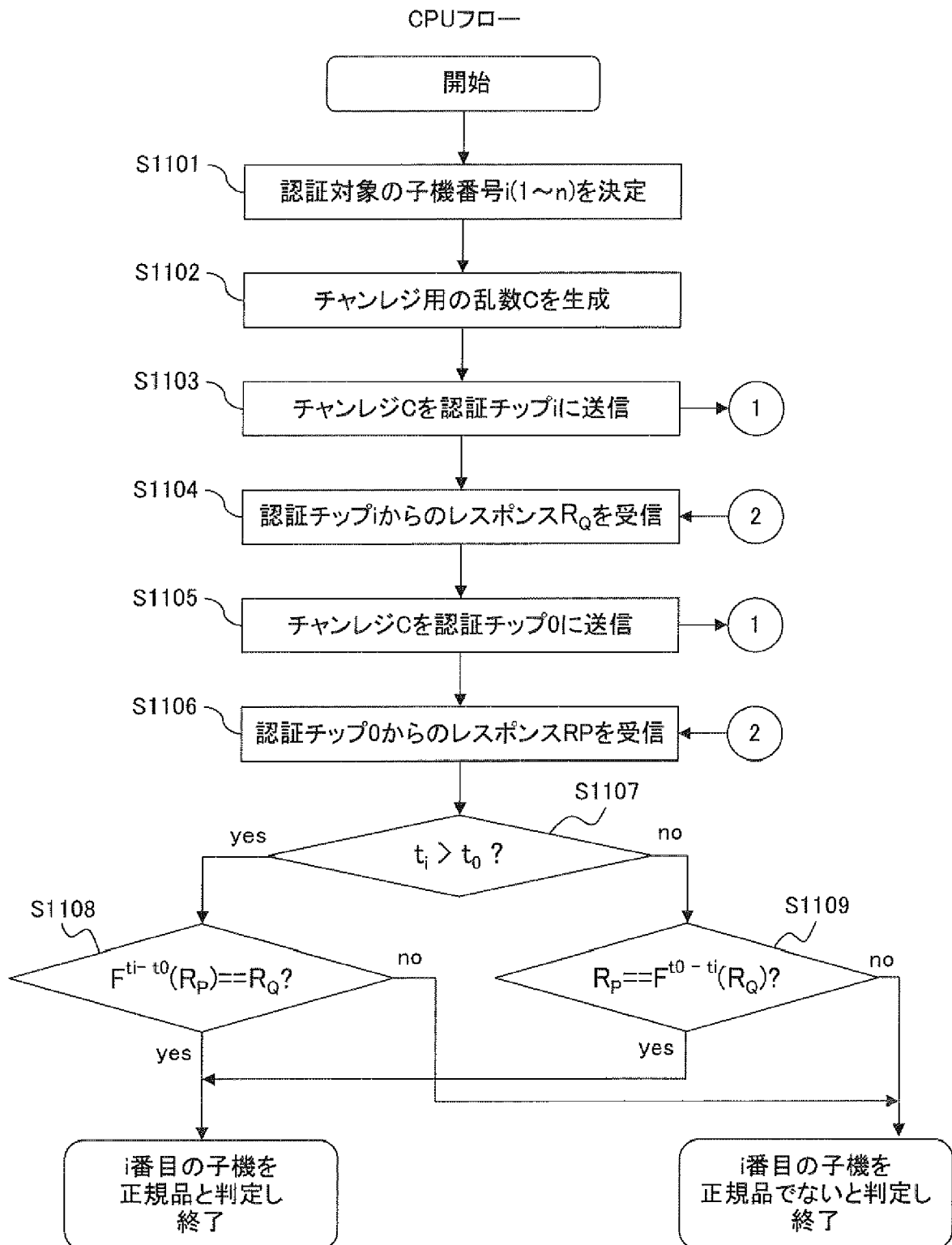
本発明：チップ内部不揮発性メモリに書かれた
バスアドレス情報をもとに、親機と子機で動作を切替



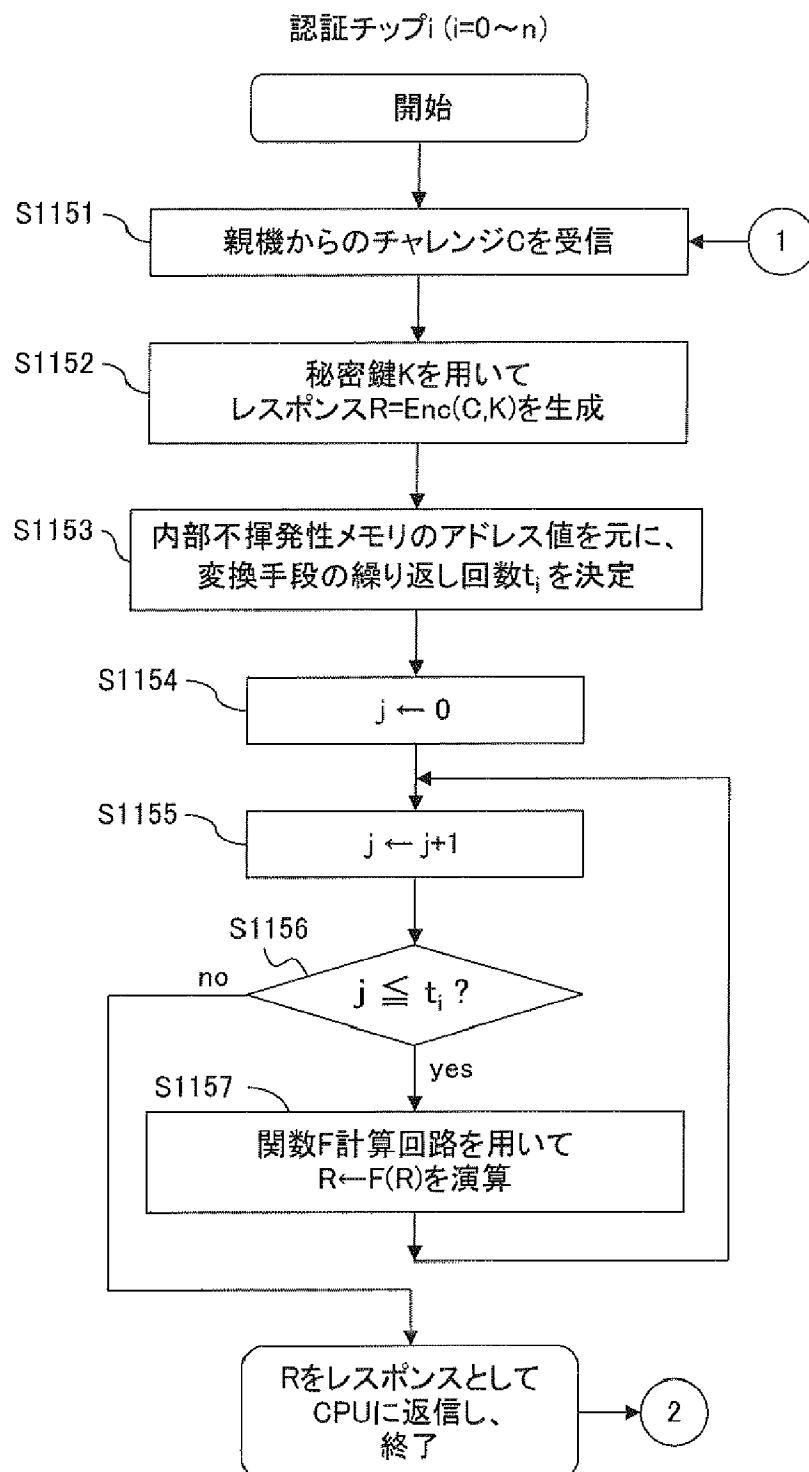
[図10]



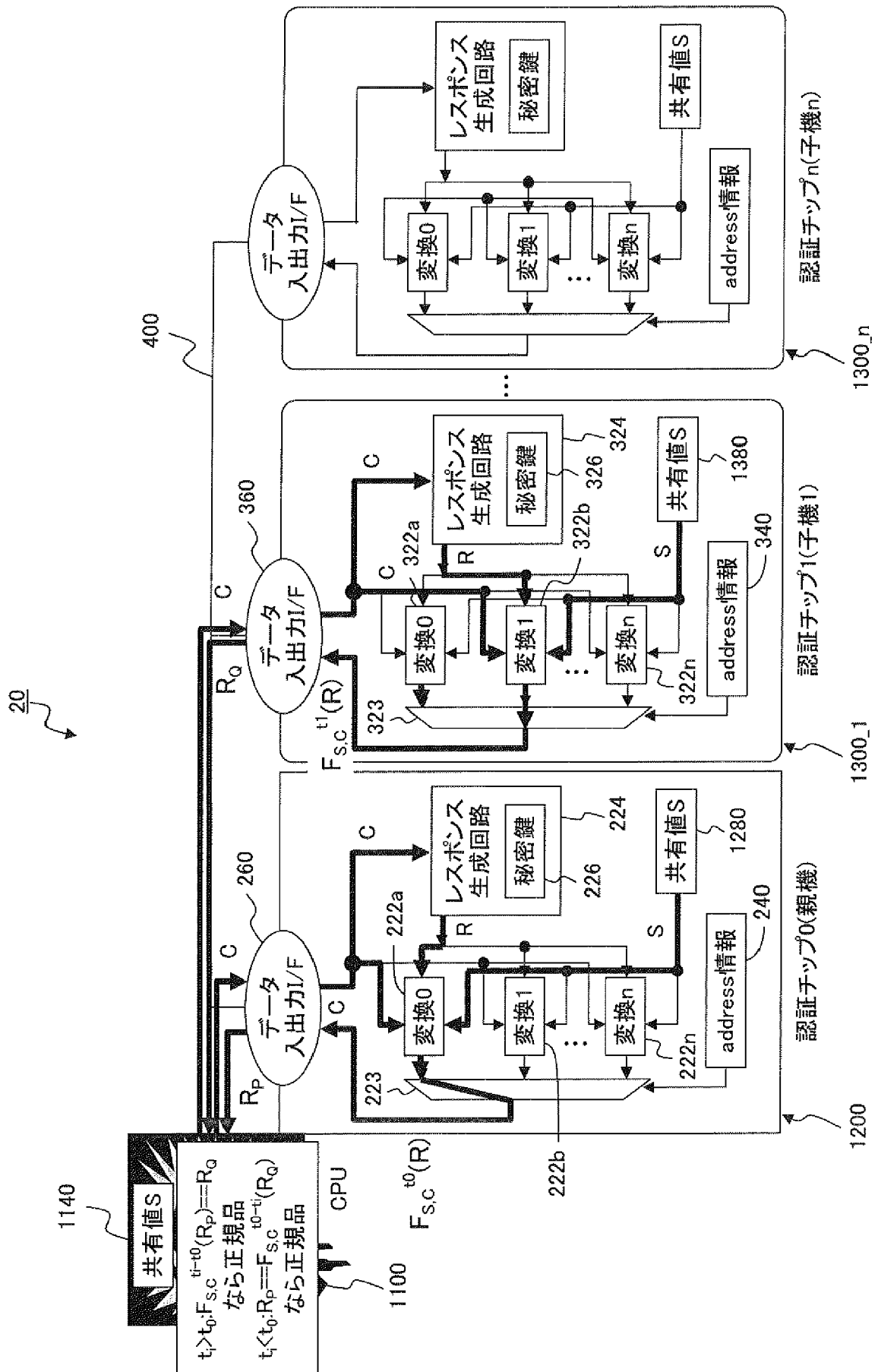
[図11]



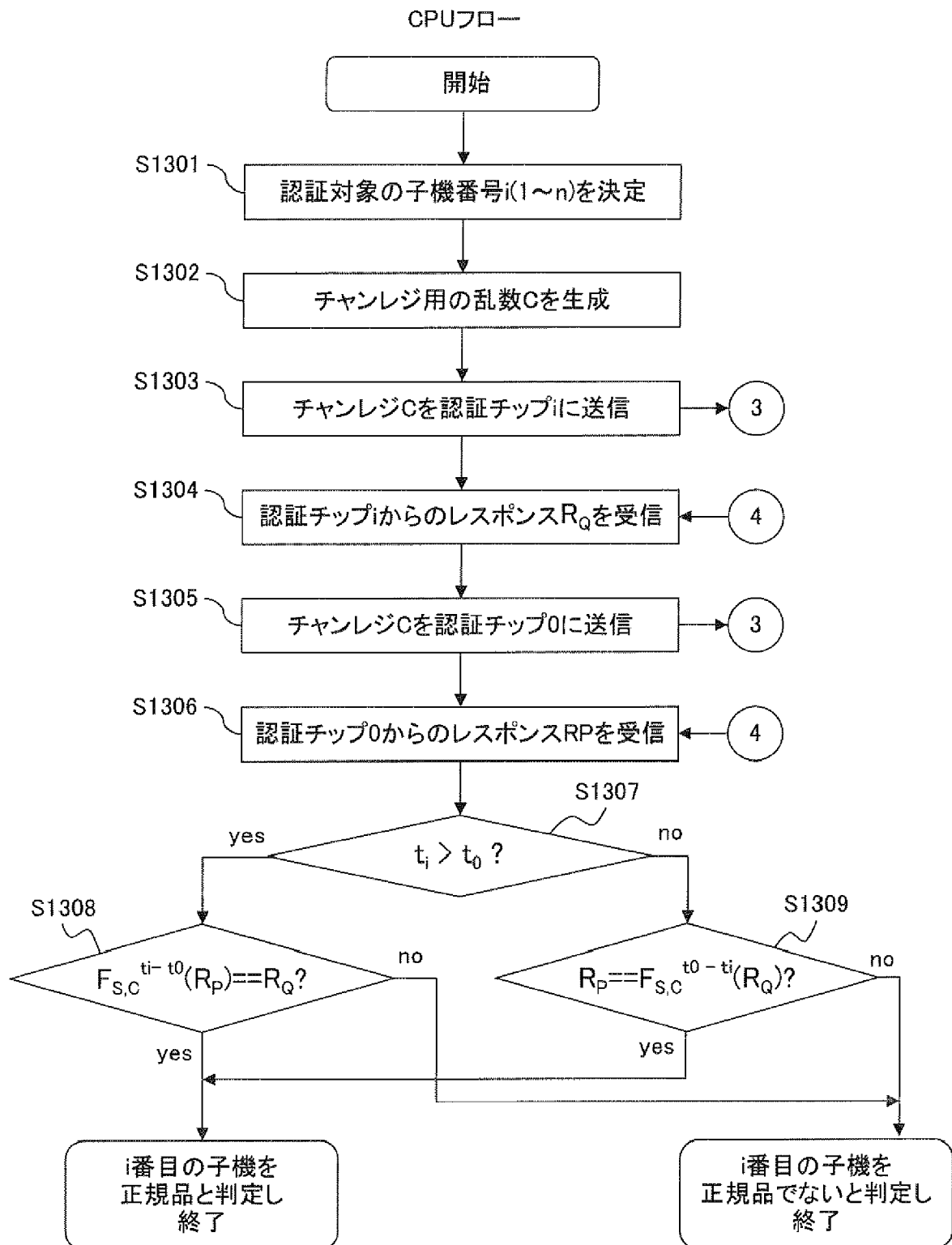
[図12]



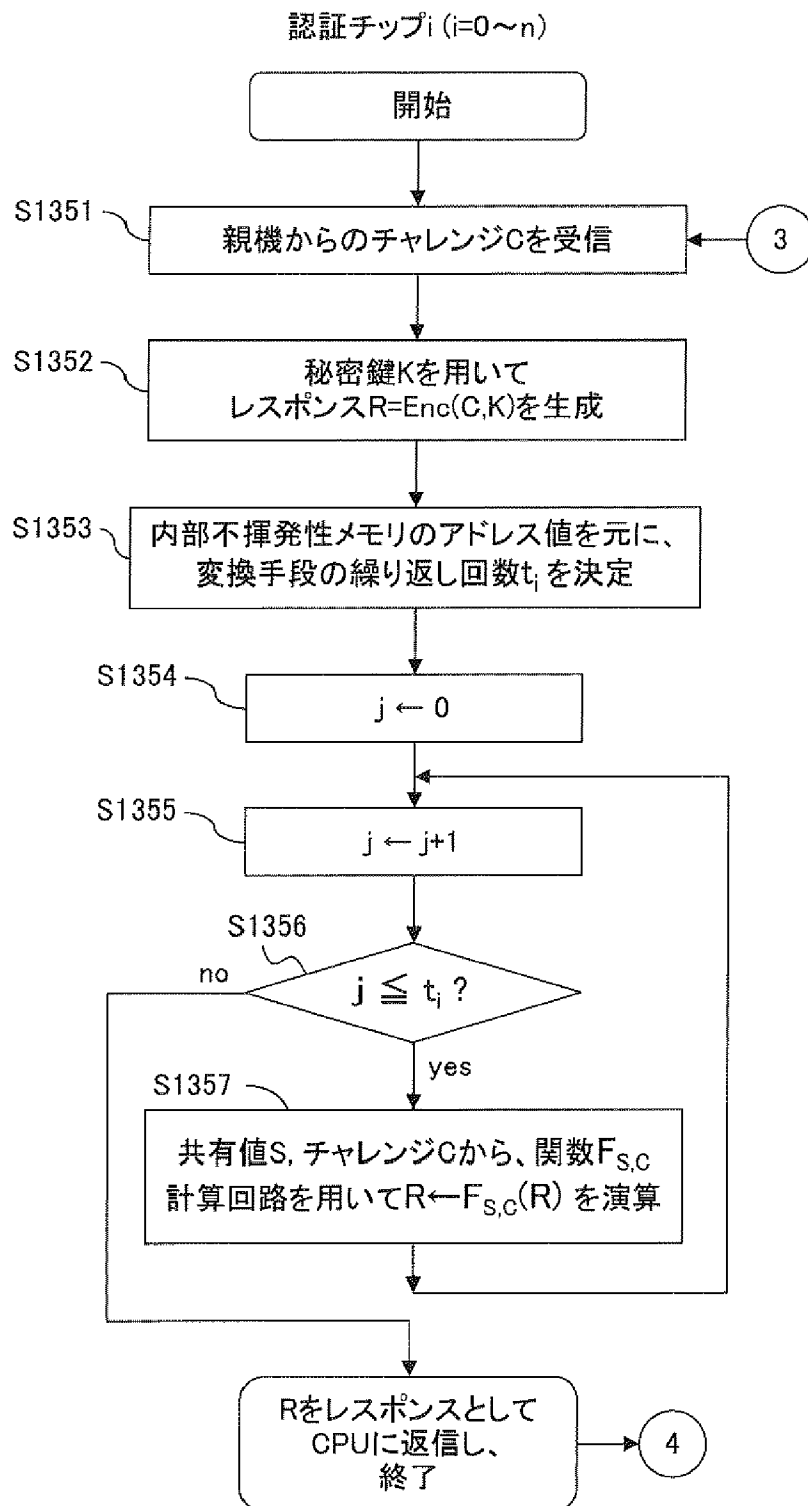
[図13]



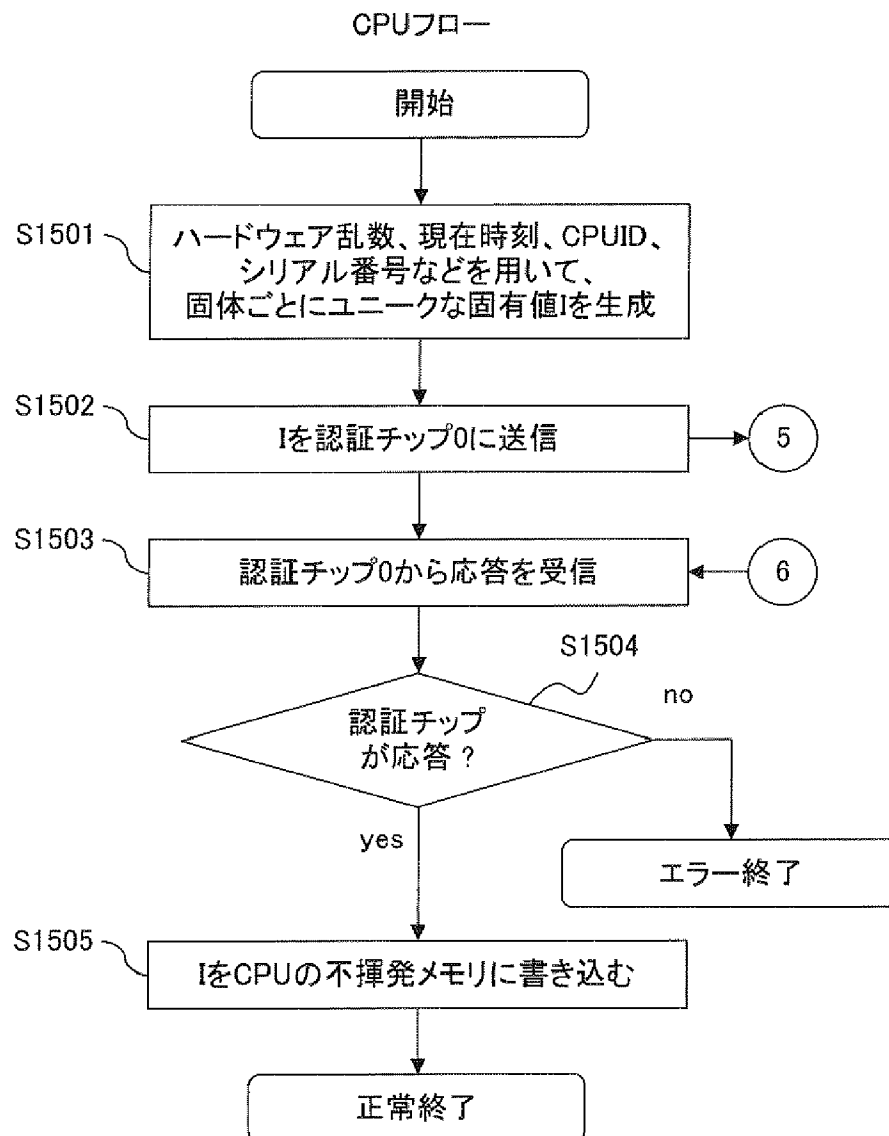
[図14]



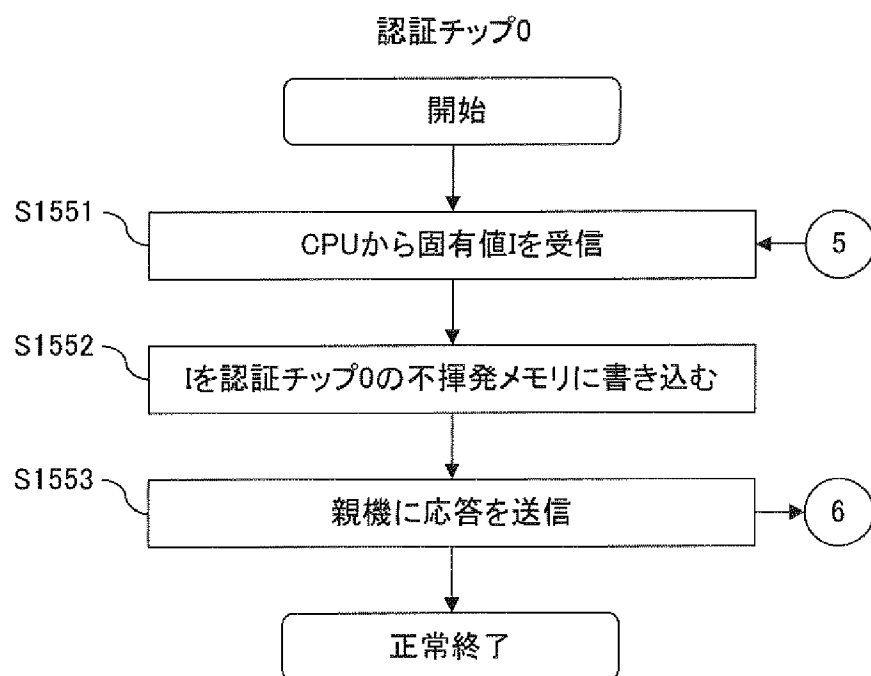
[図15]



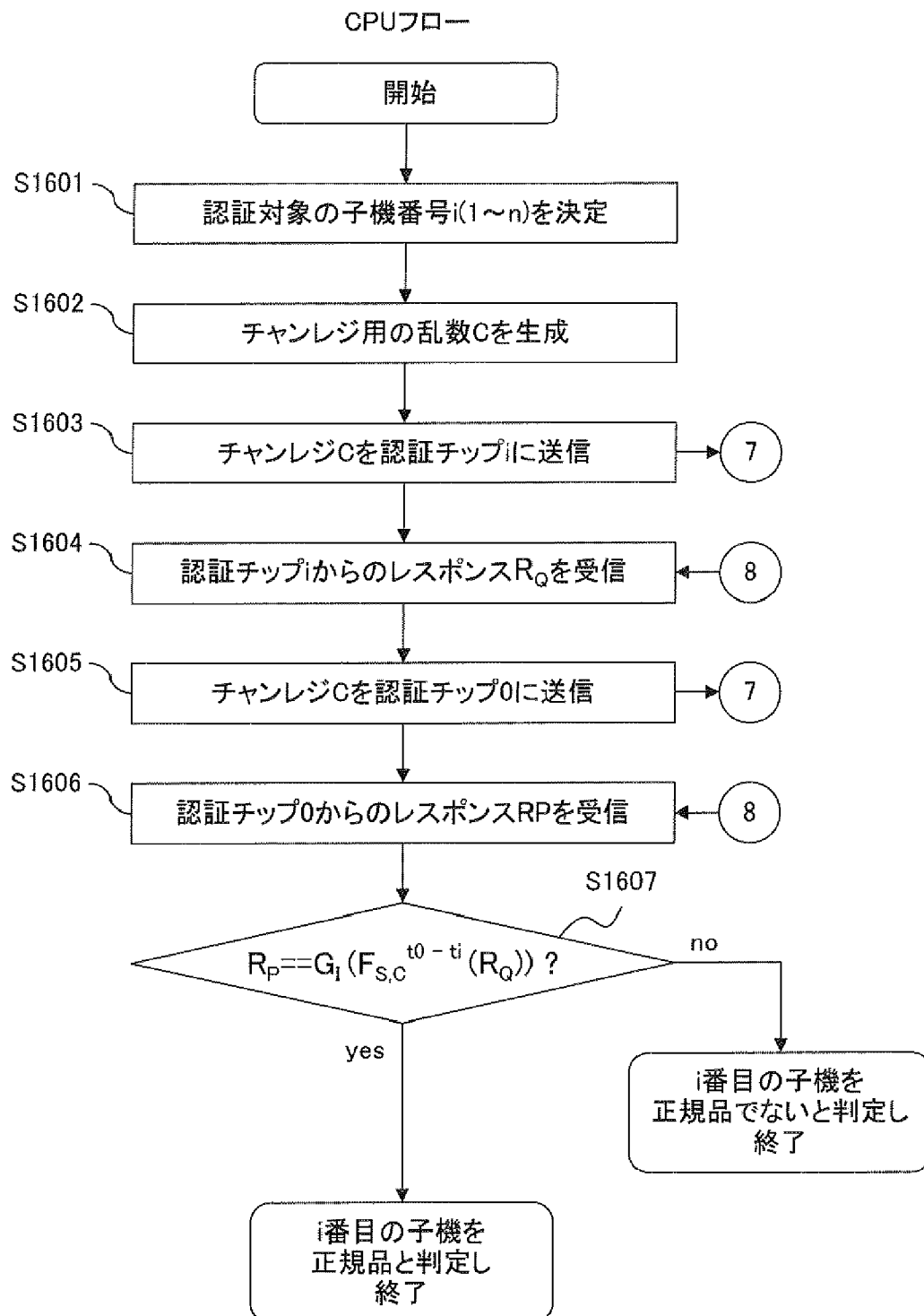
[図17]



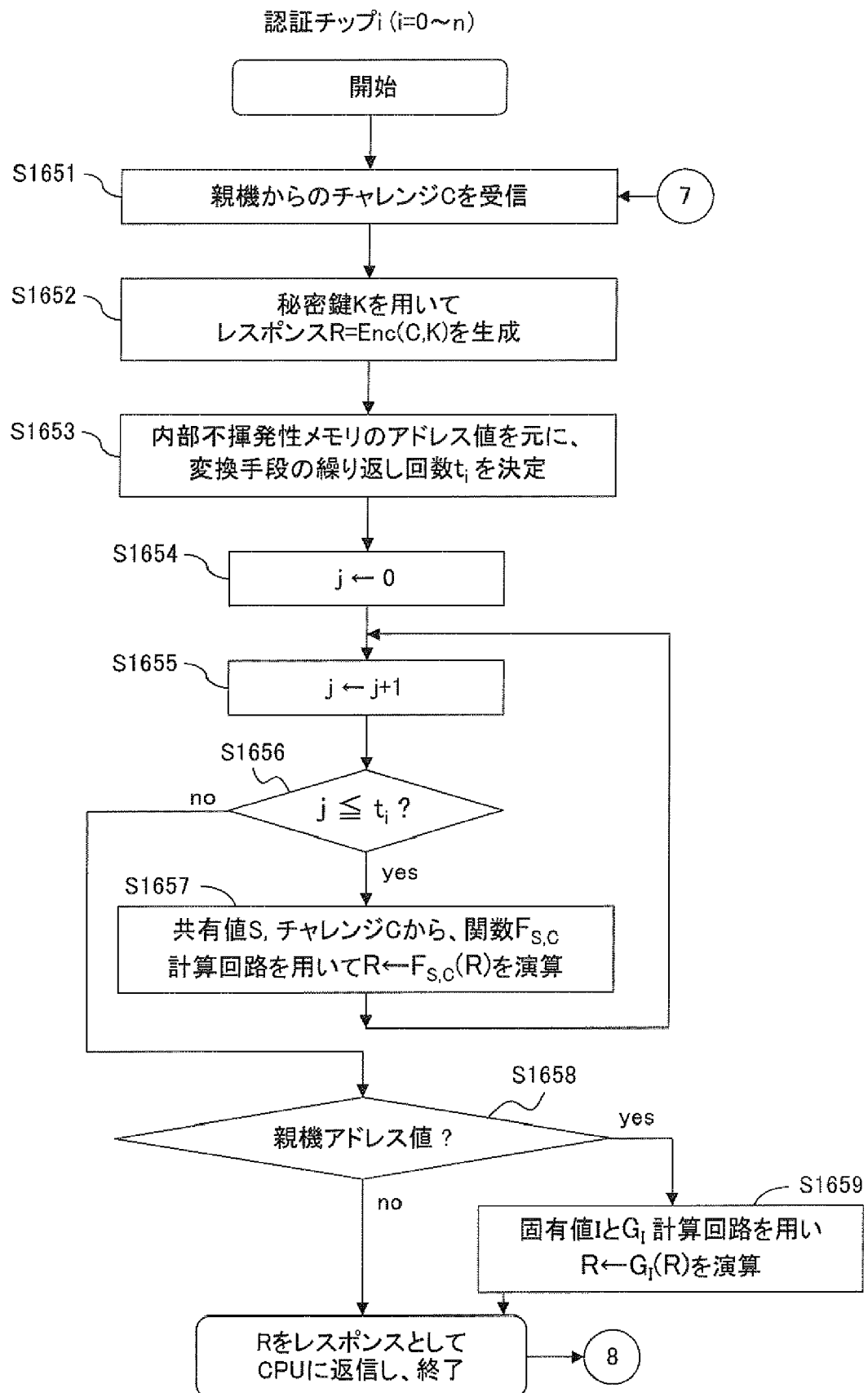
[図18]



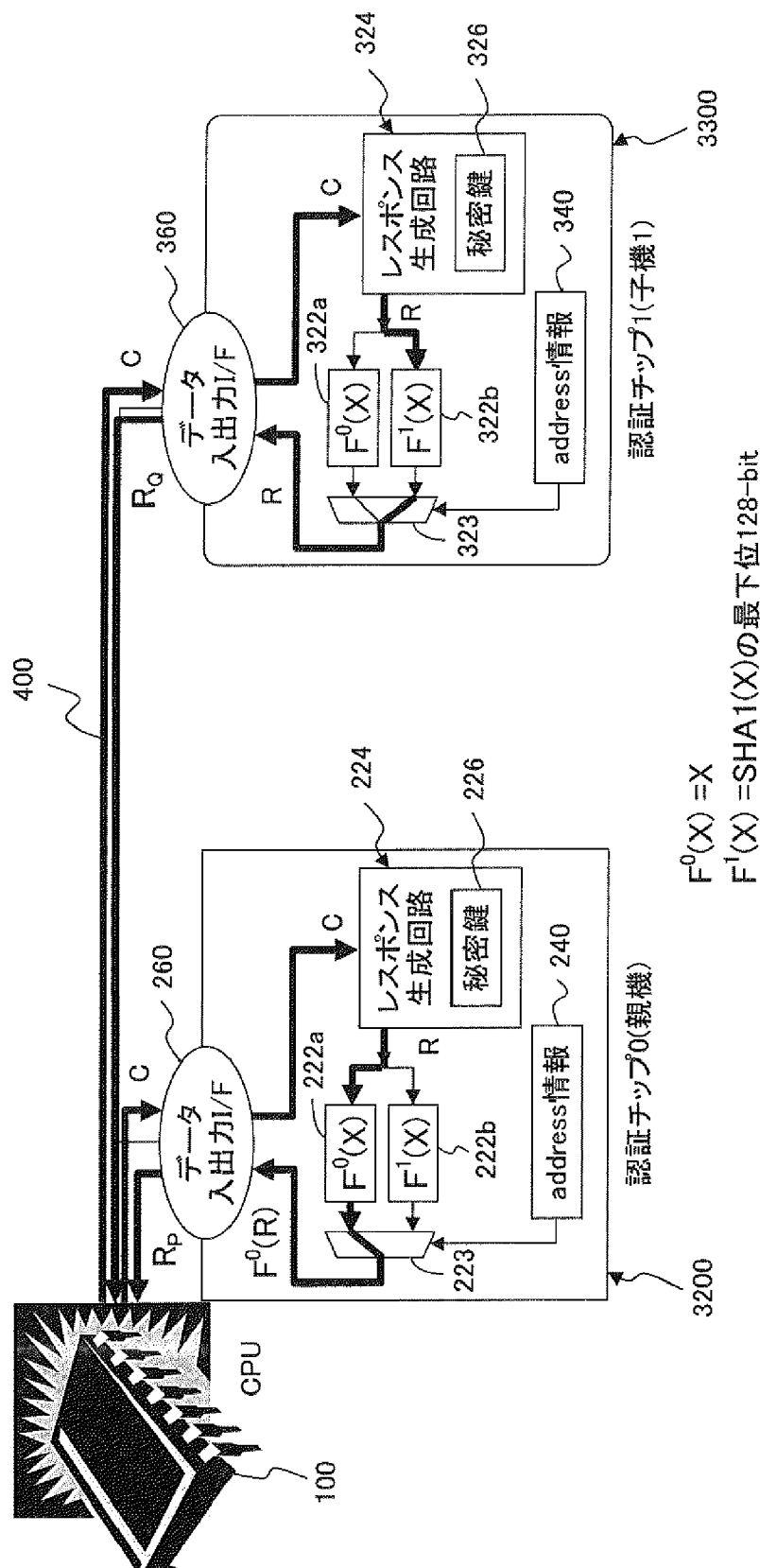
[図19]



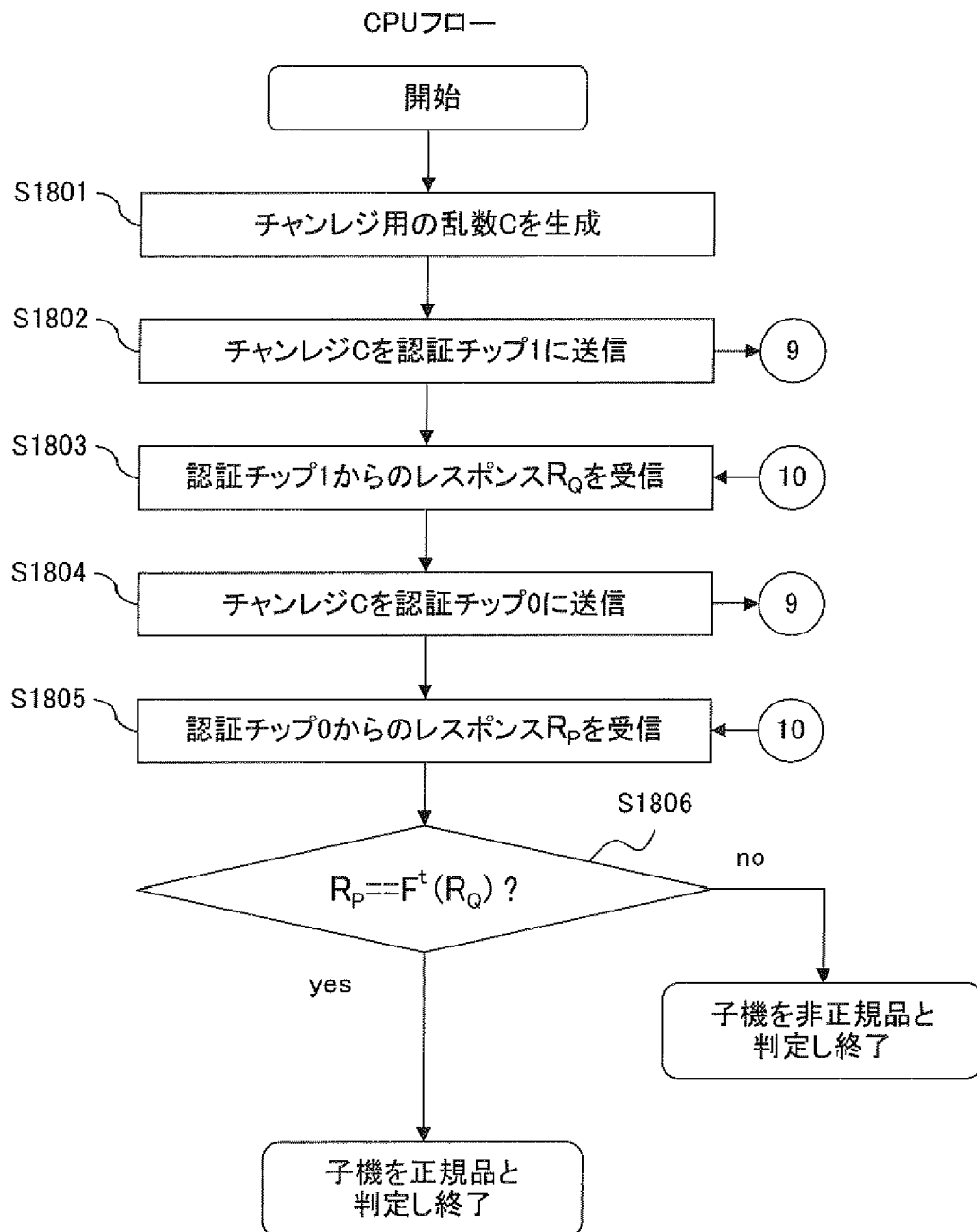
[図20]



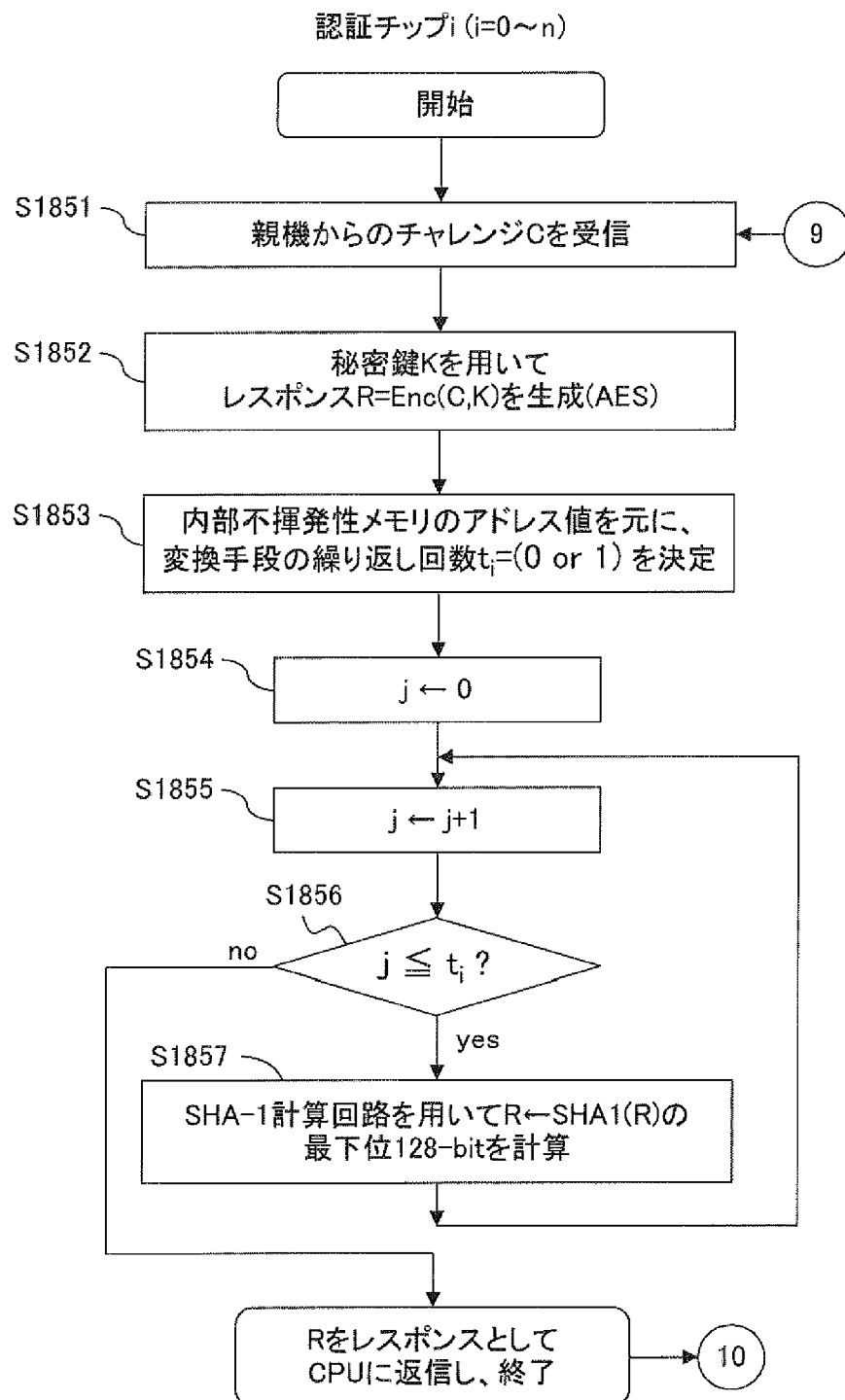
[図21]



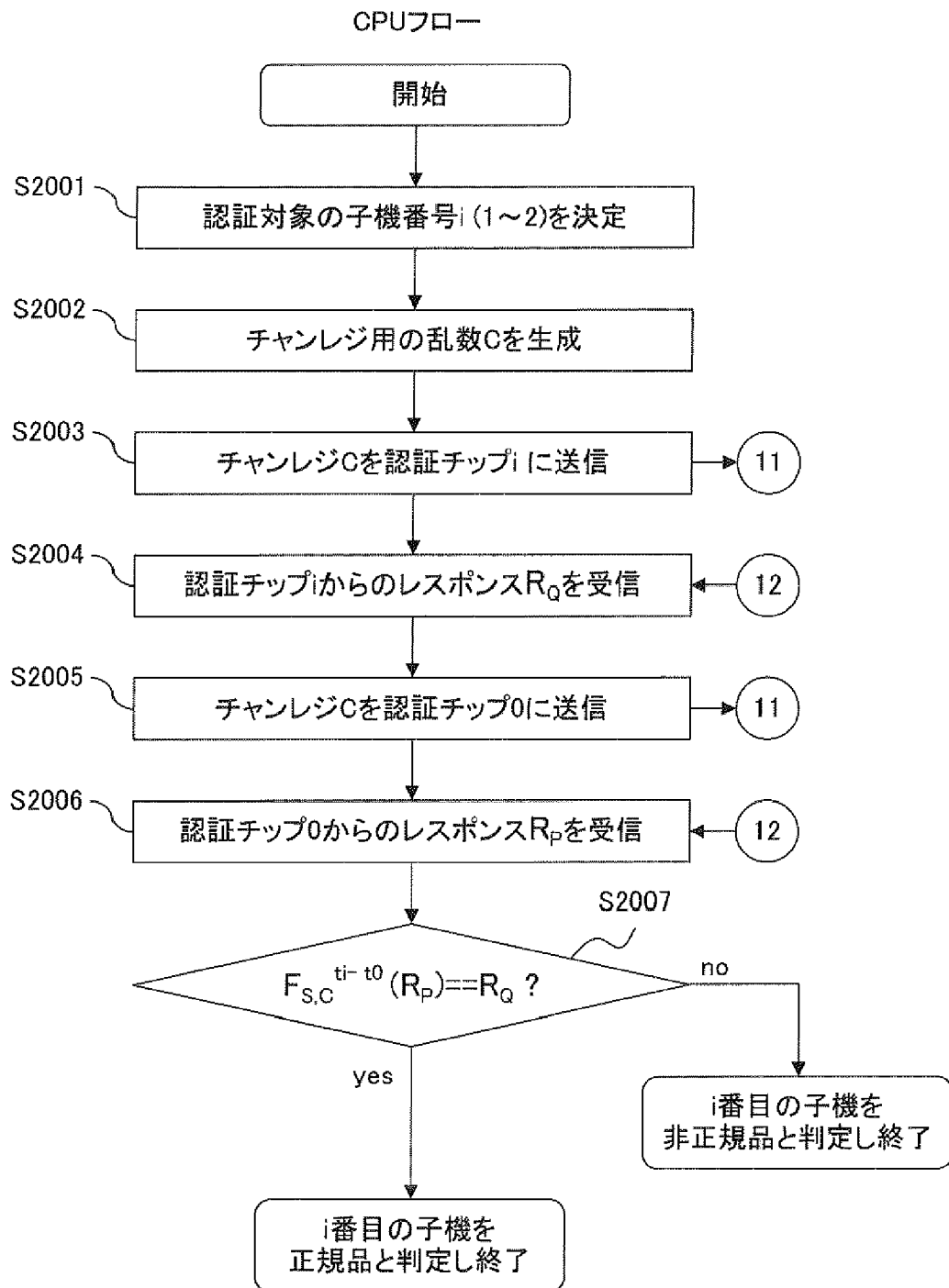
[図22]



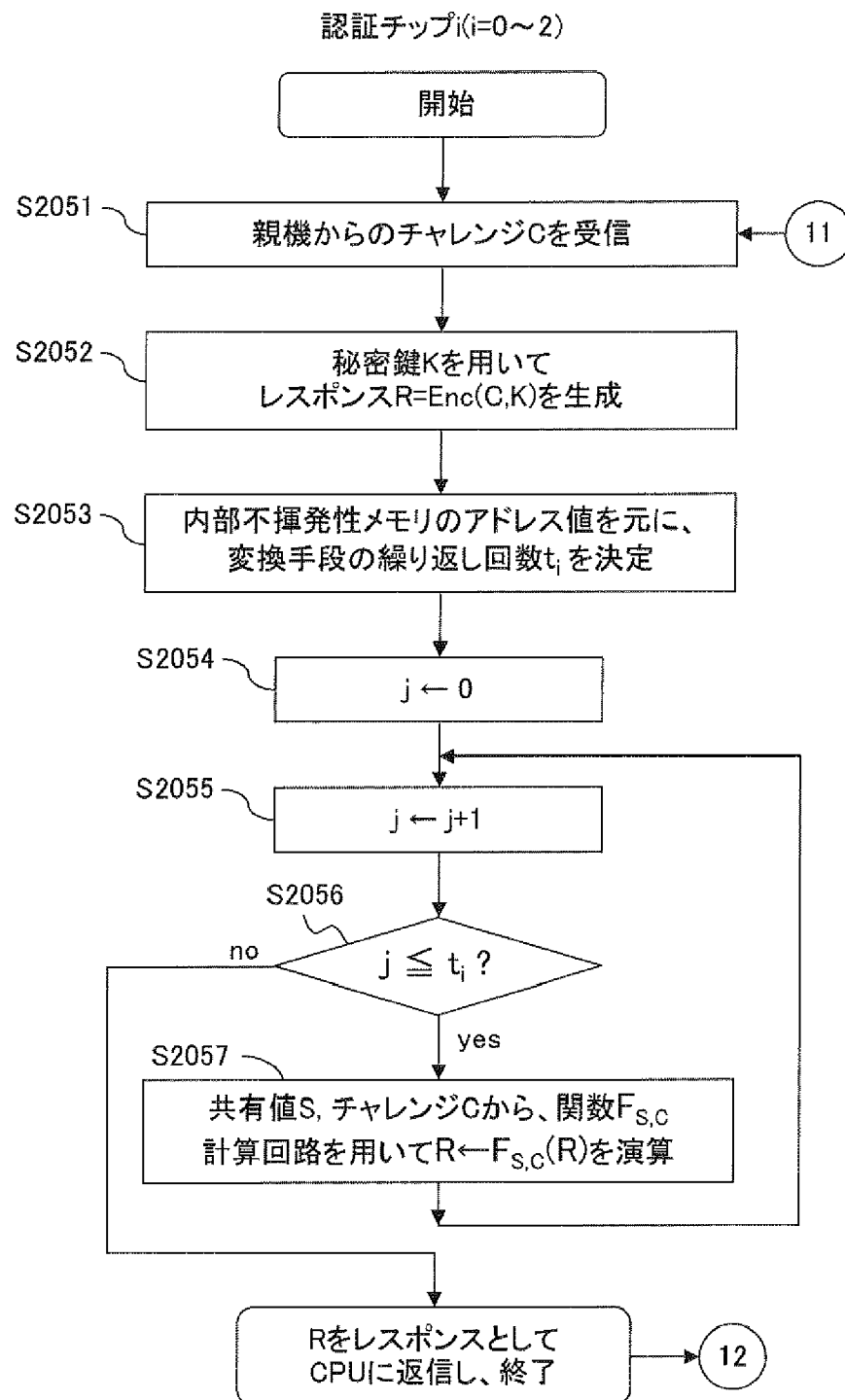
[図23]



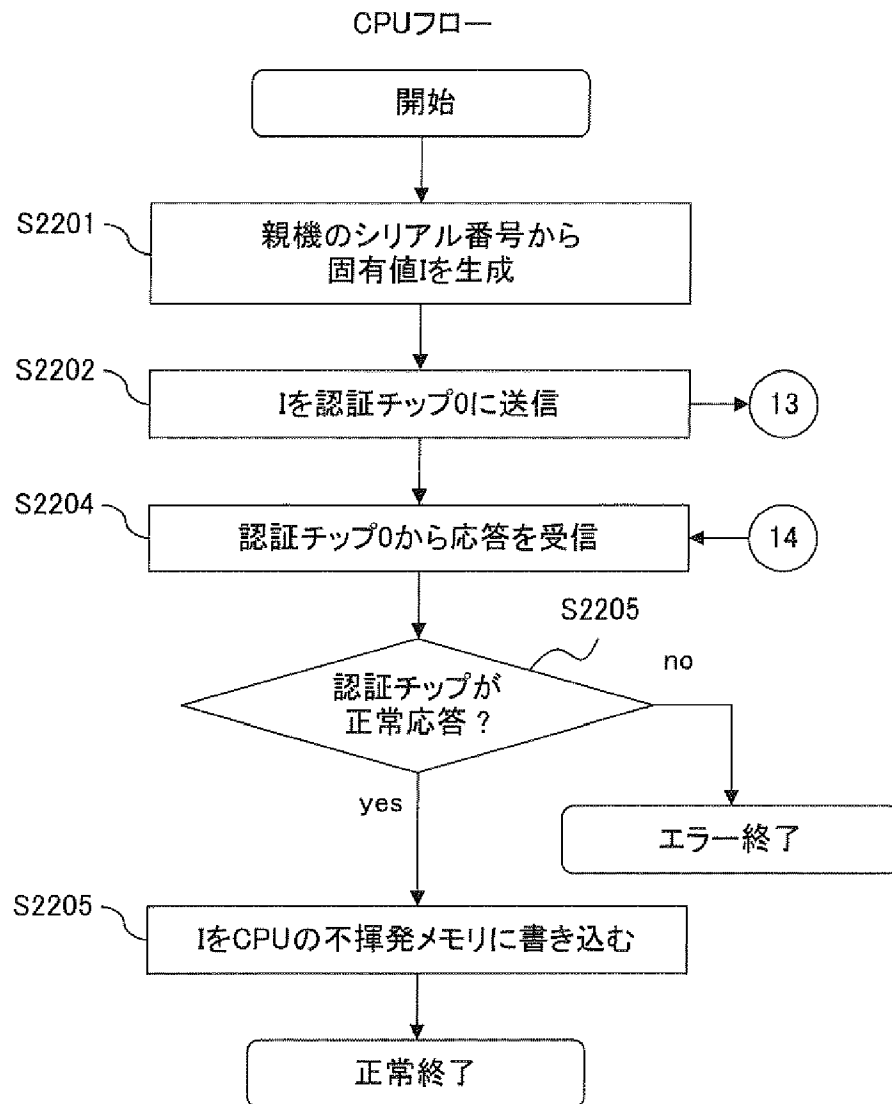
[図25]



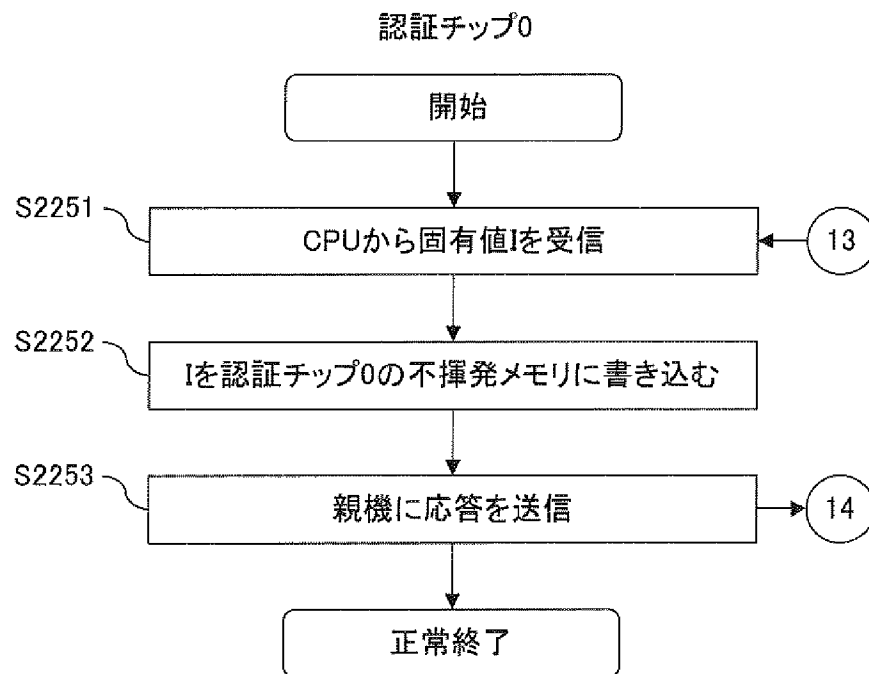
[図26]



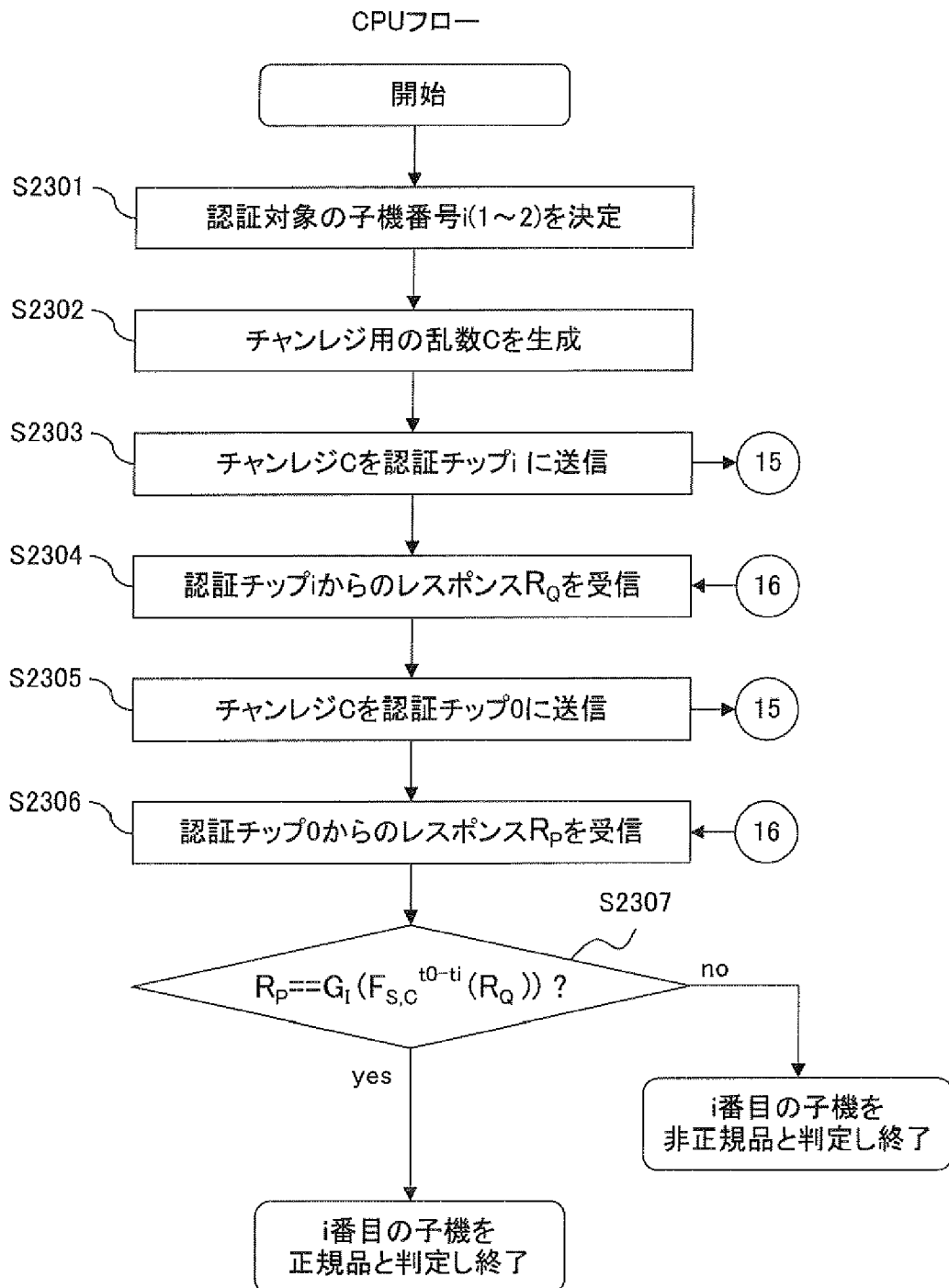
[図28]



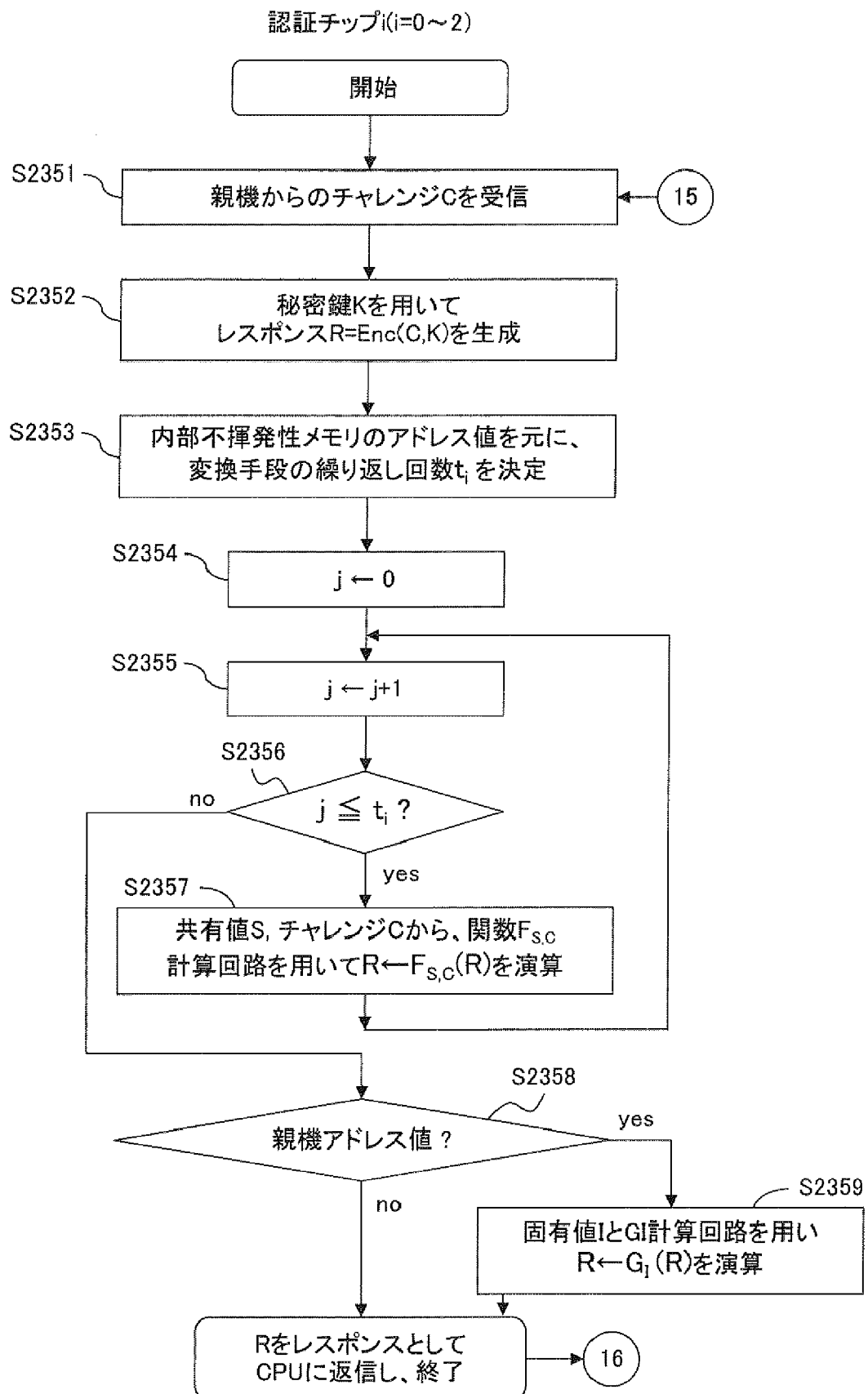
[図29]



[図30]



[図31]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/055557

A. CLASSIFICATION OF SUBJECT MATTER

H04L9/32 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L9/32

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2011
Kokai Jitsuyo Shinan Koho	1971-2011	Toroku Jitsuyo Shinan Koho	1994-2011

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	JP 6-289781 A (Fujitsu Ltd.), 18 October 1994 (18.10.1994), paragraphs [0016] to [0028] (Family: none)	21, 24 1-20, 22, 23, 25-27
Y A	JP 2006-099509 A (Felica Networks, Inc.), 13 April 2006 (13.04.2006), paragraph [0140] & US 2006/0101136 A1	21, 24 1-20, 22, 23, 25-27
A	WO 2003/107712 A1 (TELEFONAKTIEBOLAGET LM ERICSSON), 24 December 2003 (24.12.2003), page 11, line 26 to page 20, line 14 & US 2003/0233546 A1	1-27

☐

Further documents are listed in the continuation of Box C.

☐

See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search
25 March, 2011 (25.03.11)

Date of mailing of the international search report
05 April, 2011 (05.04.11)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. H04L9/32(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. H04L9/32

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 1 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 1 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 1 年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
Y A	JP 6-289781 A（富士通株式会社）1994. 10. 18, 段落【0016】～【0028】 （ファミリーなし）	21, 24 1-20, 22, 23, 25-27
Y A	JP 2006-099509 A（フェリカネットワークス株式会社）2006. 04. 13, 段落【0140】 & US 2006/0101136 A1	21, 24 1-20, 22, 23, 25-27

☒ C 欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

2 5 . 0 3 . 2 0 1 1

国際調査報告の発送日

0 5 . 0 4 . 2 0 1 1

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）

郵便番号 1 0 0 - 8 9 1 5

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

新田 亮

5 S

3 8 5 7

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 4 6

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	WO 2003/107712 A1 (TELEFONAKTIEBOLAGET LM ERICSSON) 2003. 12. 24, 第 11 頁第 26 行目～第 20 頁第 14 行目 & US 2003/0233546 A1	1-27