トラストアンカー回復技術について

海上 勇二† 前田 学† 布田 裕一†

長谷川 真吾‡ 磯辺 秀司‡ 小泉 英介‡ 酒井 正夫‡ 静谷 啓樹‡

†パナソニック株式会社
571-8501 大阪府門真市大字門真 1006 番地
unagami.yuji@jp.panasonic.com

‡東北大学 大学院情報科学研究科
980-8576 仙台市青葉区川内 41

あらまし 本稿では，機器内のセキュリティ機能の源「トラストアンカー」を機器の外部から安全かつ確実に更新する技術「トラストアンカー回復技術」を提案する．トラストアンカー回復技術では，機器内の複数の更新モジュールが連携し，更新を安全かつ確実に実行する．トラストアンカーの更新では，更新モジュール間の相互監視と多重暗号による更新により，新トラストアンカーの漏洩を防止する．

The Trust Anchor Renewal Technology

Yuji Unagami† Manabu Maeda† Yuichi Futa†

Shingo Hasegawa‡ Shuji Isobe‡ Eisuke Koizumi‡ Masao Sakai‡ Hiroki Shizuya‡

†Panasonic Corporation
1006, Oaza Kadoma, Kadoma-Shi, Osaka 571-8501, Japan
unagami.yuji@jp.panasonic.com

‡Graduate School of Information Sciences, Tohoku University
Kawauchi 41, Aoba-ku, Sendai, 980-8576, Japan

Abstract We propose the technology to renew a trust anchor, the base of security capability within a device, securely and certainly from a server external to the device. Plural install modules in a device cooperate to renew the trust anchor. At a renewal of the trust anchor, a new trust anchor is prevented from leaking by the mutual falsification detection between install modules and the renewal using a multiple encryption.

1 はじめに

近年，インターネットにつなぐことで，様々なサービスが受けられるネット家電が広がりつつある．例えばテレビにおいては，インターネットを経由して，動画コンテンツを有料配信するサービスが普及している．また，携帯電話においては，ネットバンキングなどのサービスが普及している．これらの有料コンテンツ配信やネットバンキングなどのサービスは，機器内のアプリケーション（以後，アプリ）によって，実行されている．これらのアプリは，有料コンテンツやユーザの個人情報を扱うため，不正な解析や改ざんから保護する必要がある．本稿では，機器内でこの保護のための検証プログラムや鍵を安全に保管・実行する部分を，セキュリティ機能の源，あるい

は錨という意味で，「トラストアンカー」と呼ぶ．

トラストアンカーは現在のネット家電ではハードウェア実装されていることが多い．しかし，トラストアンカーをハードウェア実装すると，万が一トラストアンカー自体が不正な解析や改ざんをされて，セキュリティ劣化した場合，ハードウェア自体を更新（交換）する必要がある．そのため，非常にコストがかかってしまう．

そこで，ここでは，ハードウェア交換よりもコストを削減した方法として，トラストアンカーをソフトウェア実装してネットワーク経由などによりリモートで更新することを考える．既に，ネット家電において，ネットワーク経由でアプリなどのソフトウェアの更新が行われている．この場合，トラストアンカーがソフトウェア更新処理を保護することによって，確実に更新していた．しかし，トラストアンカーの更新時には，セキュリティ機能の源となるモジュールが存在しないため，そのままの適用はできず新たな方式開発が必要となる．

本稿では，トラストアンカーを外部のサーバから安全かつ確実に更新する技術「トラストアンカー回復技術」を提案する．

2 節では，トラストアンカー回復技術の要件と概要を説明する．3 節では，トラストアンカー回復の全体処理及び，最も重要な回復処理の詳細について説明する．4節では，提案するトラストアンカー回復技術の考察を述べる．

2 トラストアンカー回復技術

トラストアンカー回復技術のモデルとして，図 1 で示す構成を考える．すなわち，機器内に更新モジュールを備え，これが外部のサーバから新たなトラストアンカーに更新する．ここで，機器内のトラストアンカーはすでに攻撃対象となり改ざん等が行われセキュリティ劣化しているものとする．

トラストアンカーの更新では，サーバから暗号化した新トラストアンカーを更新モジュールへ送信する．更新モジュールは，暗号化した新トラストアンカーを復号し，更新する．

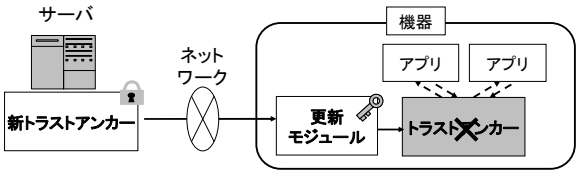

図 1 トラストアンカーの更新

2.1 トラストアンカー回復技術の要件

更新モジュールを用いてトラストアンカーを更新する場合，更新モジュールも攻撃対象になり，同時にセキュリティ劣化している可能性が高い．更新モジュールがセキュリティ劣化をしていると，トラストアンカーが確実に更新されない恐れがある．具体的には，セキュリティ劣化した更新モジュールが，トラストアンカーを更新しないようにしたり，新トラストアンカーを改ざんするようにしたりすることが考えられる．

また，改ざんされた更新モジュールから復号した平文の新トラストアンカーを不正に漏洩させることで，攻撃者の解析が可能となる．ここで，暗号化した新トラストアンカーであっても，復号鍵が漏洩した場合は，同様に攻撃者が平文の新トラストアンカーを取得できてしまう．

トラストアンカーは，難読化などの耐タンパー技術[2]により不正な解析や改ざんを困難にしている．しかし，攻撃者に解析されると，新たに施した耐タンパー技術も早期に破られる可能性が高くなる．耐タンパー技術が破られると，再度トラストアンカーを更新する必要があり，更新が頻繁になるためコストがかかってしまう．

すなわち，トラストアンカー回復技術では，以下の 2 つが要件となる（図 2）．

① 更新モジュールがセキュリティ劣化をしてもトラストアンカーが更新可能
② 新トラストアンカーの漏洩防止
    1) 平文の新トラストアンカーの漏洩防止
    2) 復号鍵と暗号化トラストアンカーが漏洩しても，平文の新トラストアンカーが取得不可能

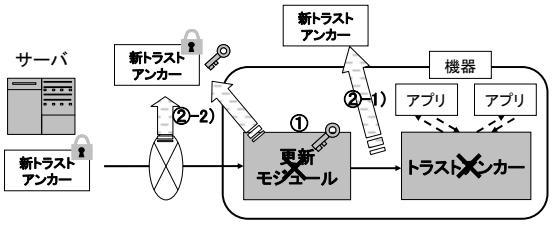

図2 トラストアンカー回復技術の要件

2.2 トラストアンカー回復技術の概要

2.1 節の要件を満たすため，機器内に複数の更新モジュールを導入し，複数の更新モジュールによって，トラストアンカーの更新を確実に実行する．(図 3)．

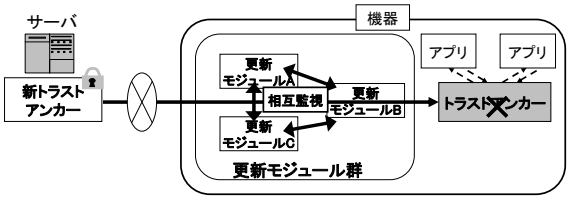

図 3 トラストアンカー回復技術

複数の更新モジュール同士では，定期あるいは不定期に相互に改ざん検出する「相互監視」を行う．これにより，更新モジュールのセキュリティ劣化を検出することができる．

トラストアンカーの更新時に，相互監視の結果を用いて，セキュリティ劣化していない更新モジュールをサーバが選択し，トラストアンカーを更新させる．よって，複数の更新モジュール全てが同時に攻撃されない限り，トラストアンカーを確実に更新できる．

また，相互監視を頻繁に行うことにより，新トラストアンカーが外部に漏洩する前に，更新モジュールのセキュリティ劣化を検出する．これにより，新トラストアンカーの漏洩を防止できる．

さらに，新トラストアンカーを多重暗号化し，一つの更新モジュールに復号処理を集中させずに，複数の更新モジュールで復号処理を行う．復号処理と相互監視処理を連係することで，全ての復号鍵の漏洩を防止でき，攻撃者が平文の新トラストアンカーを取得できない．

3 トラストアンカー回復の処理

3.1 全体の処理の流れ

トラストアンカー回復の処理は，次の 3 つの処理から成り立つ．

1. 初期設計処理

トラストアンカーを更新するために必要となる各種鍵データや，トラストアンカー更新後に必要となるデータなどを更新モジュールに埋め込む．データの埋め込み方法は，参考文献[1]において提案されている．

2. 検知処理

機器内の更新モジュールが，トラストアンカーの改ざん検出を実行する．改ざんを検出した場合，サーバに通知する．サーバは，トラストアンカーを更新するかどうか判断する．この判断が必要な理由は，改ざんを検出したと通知してきた更新モジュール自体が改ざんされていることも考えられるからである．そのため，サーバは，他の更新モジュールにトラストアンカーの改ざん検出を実行させる．各更新モジュールは，改ざん検出結果をサーバに送信する．サーバは，各更新モジュールの改ざん検出結果からトラストアンカーがセキュリティ劣化したか否かの判断をする．

3. 回復処理

複数の更新モジュール間で相互監視を行うとともに，トラストアンカーを更新する．回復処理の詳細は，3.2 節で述べる．

上記 3 つの処理の流れとして，まず，機器製造時に初期設計処理を実行する．初期設計処理の完了後，出荷され，ユーザに利用される．

ユーザが機器を利用している際には，トラストアンカーがアプリを攻撃者から保護する．これと同時に，各更新モジュールがトラストアンカーの改ざん検出を行う検知処理を実行する．

検知処理を行った結果，トラストアンカーが改ざんされたと判明し，サーバがトラストアンカーの更新が必要と判断した場合には，回復処理へ

移行する.

回復処理の完了後, 再び検知処理へ戻る.

3.2 回復処理

本節では, トラストアンカー回復技術の処理で最も重要である回復処理の詳細を説明する.

トラストアンカー回復技術の要件への対策として, (1)相互監視処理, (2)多重暗号による更新処理について説明する.

3.2.1 相互監視処理

相互監視処理では, 複数の更新モジュール同士で, 各更新モジュールが他の更新モジュールに対して, 改ざん検出の処理を行う.

改ざん検出は, 更新モジュールのハッシュ値を計算し, 初期設計処理で予め計算されて, 保持しているハッシュ値と比較する. 参照したハッシュ値と一致するか否かで改ざんされたか否かを判定する. どの更新モジュールを監視するかも, 初期設計処理で予め設定されている.

各更新モジュールは, 改ざん検出結果をサーバへ通知する. サーバは各更新モジュールから受信した改ざん検出結果から, 改ざんされた更新モジュールがあるか否かを判定する.

相互監視処理は, 新トラストアンカーの復号処理中に定期的に実行される. 定期的に実行する時間間隔は, 例えば, 新トラストアンカーの解析できないほどの一部のみしか, 通信路を通して外部に出力されないように, 短い間隔を設定する.

よって, 相互監視処理が頻繁に実行されることで, 改ざんされた更新モジュールを即座に検出し, 平文のトラストアンカーの漏洩を防止することができる.

3.2.2 多重暗号による更新処理

多重暗号による更新処理では, サーバが新トラストアンカーを複数の鍵を用いて多重に暗号化し, 更新を担当する複数の更新モジュールに送信する. 複数の更新モジュールは, 暗号化された新トラストアンカーを復号しながら, 更新する(図 4). このとき, 多重に暗号化された新トラストアンカーを復号するための複数の鍵を, 各更新モジュールへ送信するタイミングをサーバが制御する. このことにより, 攻撃者が復号鍵を入手することを困難にする.

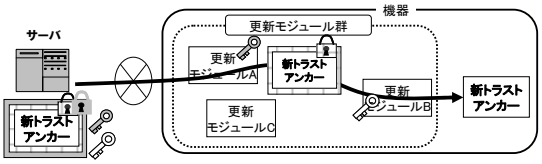

図 4 多重暗号による更新

具体的な更新処理について説明する. ここでは, 二重に暗号化する例を用いて説明する.

i) サーバは, 多重に暗号化するための暗号鍵を生成する. ここでは, 第 1 の鍵及び第 2 の鍵の 2 つを生成する.

ii) サーバは, 第 2 の鍵で新トラストアンカーを暗号化し, 暗号化トラストアンカーを生成する.

iii) サーバは, 第 1 の鍵で暗号化トラストアンカーを暗号化し, 多重暗号化トラストアンカーを生成する.

iv) サーバは, 複数の更新モジュールから任意の更新モジュール(更新モジュール A)を一つ選択する. 選択した更新モジュール A に多重暗号化トラストアンカーを送信する. さらに, 更新モジュール A に第 1 の鍵を送信する.

v) 更新モジュール A は, 第 1 の鍵を用いて, 多重暗号化トラストアンカーを復号し, 暗号化トラストアンカーを取得する. 復号が終了するとサーバへ通知する.

vi) サーバは, 複数の更新モジュールから更新モジュール A とは異なる更新モジュール(更新モジュール B)を一つ選択する. 選択した更新モジュール B に第 2 の鍵を送信する. また, 更新モジュール A に対して, 更新モジュール B へ暗号化トラストアンカーを送信するよう依頼する.

vii) 更新モジュール B は, 第 2 の鍵を用いて,

暗号化トラストアンカーを復号し，新トラストアンカーを取得する．

viii) 更新モジュール B は，トラストアンカーから新トラストアンカーへ上書きする．上書きが完了したらサーバへ通知する．

ix) 各更新モジュールは，サーバから受信した検証用の証明書を用いて，新トラストアンカーが正しく更新されたかを検証する．

上記の更新処理により，一つの更新モジュールに復号処理が集中することがなくなる．そのため，多重に暗号化した新トラストアンカーを復号するためには，複数の更新モジュールを攻撃する必要がある．

3.2.3 相互監視処理と多重暗号による更新処理の連係

相互監視処理と多重暗号による更新処理の連係することで，復号鍵の漏洩を防止する．2 つの処理の連係動作を，図 5 を用いて説明する．

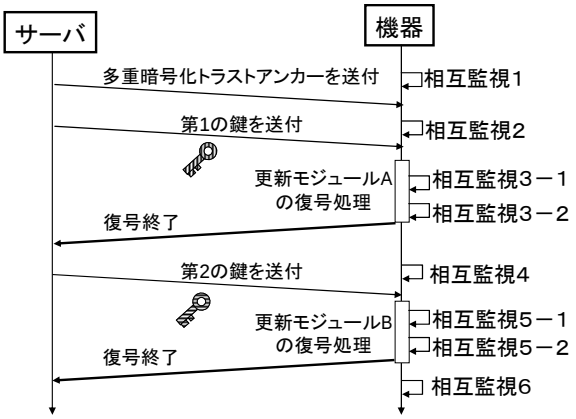

図 5 相互監視処理と多重暗号による更新処理の関係

i) 回復処理の開始後，サーバから機器へ多重暗号化トラストアンカーを送信する前に，相互監視処理（相互監視 1）を実行する．

ii) サーバから機器内の更新モジュール A へ第 1 の鍵を送信する．更新モジュール A が第 1 の鍵を受信する前に，相互監視処理（相互監視 2）を実行する．

iii) 更新モジュール A が第 1 の鍵を受信し，第 1 の鍵を用いて，多重暗号化トラストアンカーを復号する．このとき，復号が完了する前に，定期的に相互監視処理（相互監視 3-1, 3-2）を実行する．

iv) サーバから機器内の更新モジュール B へ第 2 の鍵を送信する．更新モジュール B が第 2 の鍵を受信する前に，相互監視処理（相互監視 4）を実行する．

v) 更新モジュール B が第 2 の鍵を受信し，第 2 の鍵を用いて，暗号化トラストアンカーを復号する．このとき，復号が完了する前に，定期的に相互監視処理（相互監視 5-1, 5-2）を実行する．

vi) 各更新モジュールの新トラストアンカーの更新検証完了後に，相互監視処理（相互監視 6）を実行する．

相互監視処理では，各更新モジュールの改ざん検出結果を，サーバへ送信する．サーバは改ざん検出結果に基づいて，いずれかの更新モジュールが改ざんされているか否かを判定する．トラストアンカーの更新を担当する更新モジュールが改ざんされている場合，サーバは回復処理を停止する．

4 考察

各更新モジュールによる新トラストアンカーの更新検証完了後に，相互監視処理を行い，トラストアンカーの更新を担当した更新モジュールが改ざんされていないことを確認することで，確実に更新されたことを保証できる．これにより，要件①を満たす．

相互監視処理を頻繁に実行することで，平文のトラストアンカーの漏洩を一部のみで防止することができる．これにより．要件②-1)を満たす．

また，相互監視処理と多重暗号による更新処理を連係させることで，第 1 の鍵あるいは，第 2 の鍵の送信前に，攻撃者による攻撃が検出されれば，攻撃者は多重暗号化トラストアンカーを復号するための鍵を入手することができない．こ

れにより，要件②-2)を満たす．

したがって，3 節のトラストアンカー回復の処理は，トラストアンカー回復技術のすべての要件を満たす．

5　まとめ

本稿では，トラストアンカーを外部のサーバから安全かつ確実に更新するトラストアンカー回復技術を提案した．

トラストアンカー回復技術では，複数の更新モジュールを導入し，相互監視処理と多重暗号による更新処理を連係して行うことで，更新モジュールがセキュリティ劣化をしてもトラストアンカーが更新可能となり，平文の新トラストアンカーの漏洩を防止することができる．

今後は，評価用プロトタイプを作成し，シミュレーションにより，機器内の更新モジュールの最適な数などの評価を行う．

参考文献

[1] 吉田正樹，長谷川真吾，磯辺秀司，小泉英介，酒井正夫，静谷啓樹，布田裕一，前田学，“公開鍵基盤の頑健化について，” 信学技報，ISEC2006-160，pp119-124，2006

[2] 石間宏之，齊藤和雄，亀井光久，申吉浩，“ソフトウェアの耐タンパー技術，” 富士ゼロックス　テクニカルレポート，No.13，　pp20-28，2000