軽量暗号アルゴリズム記述のための領域特化言語 Hermes の適応性評価

2018SE088　高見雄大
指導教員：横山哲郎

1　はじめに

Hermes とは軽量暗号アルゴリズム記述のための領域特化可逆プログラミング言語である．領域特化言語とは，特定の領域の問題を解決させる目的で設計された言語のことである．他にも SQL や BNF 記法も領域特化言語である．可逆プログラミング言語は uncall することで，全ての変数や配列の情報は消去する前に 0 にクリアされる．よって，暗号アルゴリズムの実装で記憶領域に情報を残すことなく実行することが出来る．したがって，記憶領域に対する攻撃から守られることが保証されている．しかし，サイドチャネル攻撃はメモリに関する対策だけでは完全な対策にはならない．従来の可逆言語 Janus では入力値に暗号処理時間が依存している．よって，サイドチャネル攻撃の一種であるタイミングベース攻撃について脆弱性がある．Hermes は Janus の良いところを残しつつ，入力値と実行時間を独立させることでタイミング攻撃から守られている．非可逆言語は暗号化プロシージャと復号プロシージャの 2 種類作成する必要性があるが，可逆な言語は暗号化と復号を一つのプロシージャで実行することが出来る．よって，非可逆な言語の様に暗号化と復号それぞれプログラムを作る必要がない．Hermes で暗号アルゴリズムをコンパイルすることにより，全ての実行が一意に定まる C のプログラムが出力されるように設計されている．しかし，Herme は開発途中の言語であり，[1] ではいくつかの暗号アルゴリズムに対して適応していることが示されている．それ以外の軽量暗号アルゴリズムについて記述することが可能であるか，仮に記述出来たとして脆弱性が無いか，適応性が証明されていない．本研究では，可逆プログラミング言語 Hermes の適応性評価を目的とする．

2　暗号

暗号とは第三者に知られることなく情報をやり取りするための手法である．暗号化する前の文を平文，暗号化後の文を暗号文と呼び，暗号文は可読でない特徴を持つ．暗号文を平文に復元することを復号と呼ぶ．平文を暗号文に書き換えるには暗号アルゴリズムを使用する．アルゴリズム [2] とは複雑な問題を解決する手法である．暗号アルゴリズムにはブロック暗号とストリーム暗号の 2 種類ある．暗号はパブリックで強い暗号アルゴリズムに鍵を使用することで，実行することが出来る．

2.1　軽量暗号

軽量暗号とは一般の暗号と比べ攻撃からの強度が弱いのではなく，限られたリソースのデバイスでも実装可能かつ安全性が高い暗号アルゴリズムのことである．限られたリソースのデバイスの性能指標は 4 つある．例での説明：人体に埋め込むタイプの精密機器は消費電力量に対する制約，RFID など回路規模に対する条件が厳しい制約，車載機器など暗号化復号する際にリアルタイム性が要求される制約，低価格の家電製品などメモリサイズが少ないメモしサイズの制約.

3　可逆プログラミング言語

可逆プログラミング言語とは全てのプログラム実行過程が可逆になるように設計されたプログラミング言語である．よって，全てのプログラムにおいて可逆性を保証している．可逆なプログラムとは，実行過程において実行直前と実行後において状態が一意に定まるプログラムのとこである．可逆なプログラムは実行過程で破壊的な変数の更新は行わず，情報を失うことはない．しかし，可逆性を保証する代わりに制約が課せられる．

4　サイドチャネル攻撃

サイドチャネル攻撃とは攻撃手法の一種である．サイドチャネル攻撃の特徴は安全性が計算量によって保証された暗号アルゴリズムでもアルゴリズムとは関係のない物理的な情報を利用することにより，暗号鍵を特定する解読手法である．例えば, 暗号化復号の処理時間を計測することや, 消費される電気量を測定，コンピュータの暗号化が実装の際に発するノイズを測定するなどが上げられる．

4.1　タイミングベース攻撃

タイミングベース攻撃とはサイドチャネル攻撃の一種である．処理時間を測定することで暗号鍵を不正に特定する手法で，タイミングベース攻撃から守るためには暗号化復号する際，暗号鍵に関係なく処理時間を独立させることでタイミングベース攻撃から守ることができる．janus は暗号化復号の際，暗号鍵に処理時間が依存してしまう．その為，タイミングベース攻撃から守ることが出来なかった．だから，新しい可逆プログラミング言語 Hermes が作られた．

5　プログラミング言語

高水準のプログラミング言語を作成するために、プログラミング言語に適切な構文構造やプログラム意味論を定義し、プログラミング言語をコンピュータ上で実行可能な実装方法を構築しなければならない．

プログラミング言語を定義するための構文構造は，私たちがプログラムを記述する際に、入力されたデータがどのような構造を持つかを定義しなければならない．これをプログラミング言語の構文論という．そして、構文論は文法構造と型システムの定義が必要である．

コンピュータがプログラムを実行するためには、プログラミング言語には文法構造が一意に定まる必要があり、曖昧な文を許していない．これらの条件を満たす文脈自由文法を用いて文法構造が定義される．

意味論文法構造 [3] は決められている文字列をプログラムとして認識する為の規則だけである．そのため、プログラミング言語を正確に定義するためには、構文構造にプログラムの計算方法を表現する、厳密な規則を定義する必要がある．

構文論を定義するには文脈自由文法に沿って構文構造のルールを定義したのち，型システムを実装し，整合性制約を加えた 2 段階で行われる．構文構造を定義するには正しいプログラム意味が必要で，プログラム演算の表現を厳格に定義する必要がある．そしてプログラミング言語の意味を定義するための方法は主に表示的意味論，公理的意味論，操作的意味論の 3 つである．

6　関連研究

6.1　Janus

Janus[4] はプログラミング言語の一種で命令型プログラミング言語である．また，構造化プログラミング方式で記述されている．命令型プログラミング言語とはプログラムを実行する際に処理の手順を明確に記述するプログラミング言語である．明確に命令するので命令型言語という．また，宣言型言語と区別され，宣言型言語は処理の手順を記述するのではなくデータ間の関係を規定する．構造化プログラミング方式とは，プログラムを順次，分岐，反復の 3 つで記述することによって，プログラムの構造を明確にすることができる方式である．これによって後にプログラムを変更する際，手直しが簡易になるメリットがある．Janus は非可逆言語と違い可逆プログラムしか実行できないように構文規則や意味規則に制約が課せられ設計されています。そのため，全てのプログラムの可逆性を保証している．

6.2　共通鍵暗号と公開鍵暗号

暗号に使われる鍵の方式は 2 通りで，共通鍵暗号と公開鍵暗号である．共通鍵暗号とは暗号化と復号に使用される鍵が同じなので鍵は 1 つである．これに対し，公開鍵暗号は暗号化と復号で別々の公開鍵と秘密鍵の 2 つを使う．共通鍵暗号は一度鍵を共有してしまえば暗号化・復号することが容易に行うことが出来るが鍵を事前に共有する必要があるため拡張性がない特徴を持っている．公開鍵暗号は共通鍵暗号と違い事前に鍵を共有する必要がないため拡張性がある．しかし，暗号化復号の一連の処理速度は共通鍵暗号と比べると何百倍も遅い特徴を持っている．近年ではハイブリット暗号方式を採用している事も多い．ハイブリット暗号方式とはその名の通り共通鍵暗号と公開鍵暗号の良い所を組み合わせて使用する暗号方式で，共通鍵暗号の鍵を公開鍵暗号で共有し，情報は共通鍵暗号方式を使用する方式である．

7　まとめ・今後の課題

現時点での研究成果として，本研究で題材となる可逆プログラミング言語 Hermes について理解を深めた．また，プログラミング言語の構文構造の定義について理解を深めた．

今後は，文献 [1] に記載された軽量暗号 TEA,RC5,Speck128 の Hermes プログラムから暗号化, 復号の C プログラムが正しく生成されることを確認する．確認でき次第これら以外の軽量暗号について調査をする．調査した軽量暗号アルゴリズムの中から 1 つアルゴリズムを選び Hermes で記述することは可能であるか検証する．また，生成された C プログラムに脆弱性が無いかを調査する．

参考文献

[1] Mogensen, T.Æ.: Hermes: A Language for Light-Weight Encryption, *Reversible Computation* (Lanese, I. and Rawski, M., Eds.), Cham, Springer International Publishing, pp.93–110 (2020).

[2] 田辺　誠, 中島玲二, 長谷川真人：コンピュータサイエンス入門論理とプログラム意味論，Vol.6, No.4, 岩波書店 (2006).

[3] 西村　進, 大堀　淳：コンピュータサイエンス入門　アルゴリズムとプログラミング言語, *International Journal of theoretical physics*, Vol.21, No.3/4, pp.219–253 (1982).

[4] 由侑新海，秀明田中，哲郎横山：可逆プログラミング言語の引数渡し機構の拡張，情報処理学会論文誌プログラミング（PRO），Vol.7, No.4, pp.21–36 (2014).