MY-ELLTY ECMR-XXX-h 詳細評価報告書

MY-ELLTY ECMR-160/192/OEF-h は、楕円曲線が定義される体は異なるが、その他の部分は同じ署名スキームである。本報告書では主に MY-ELLTY ECMR-160-h の安全性評価について述べる。安全性の問題として、ハッシュ関数の出力が 80 ビットと非常に短いため、 Birthday Attack により 2^{40} オーダーの計算でハッシュ値の衝突を引き起こし、適応的選択メッセージ攻撃による存在的偽造が実現できる可能性が高いことがあげられる。この問題は、メッセージを回復するためにハッシュ値を削るという設計方針に起因している。自己評価書では、適応的選択メッセージ攻撃による存在的偽造不可が証明されているが、これはハッシュ関数がランダムオラクルと仮定しての結論であり、実際に使用するハッシュ関数は衝突困難性を満たしているとは言いがたく、不整合が起きている。その他、仕様の不備・誤り、不十分な記述などが散見される。

1 概要

MY-ELLTY ECMR-160-h、 MY-ELLTY ECMR-192-h、 MY-ELLTY ECMR-OEF-h は、楕円曲線演算を用いた署名スキームである。これらのスキームは、楕円曲線が定義される体が異なるだけであり、その他の部分は同じ署名スキームである。本報告書では、主に MY-ELLTY ECMR-160-h に関する安全性評価を述べるが、他の二つにもほとんど同じことが言える。安全性に関する問題点 (2節) および仕様書に関する問題点 (3節) を以下に指摘する。

安全性に関する問題点としては、適応的選択メッセージ攻撃による存在的偽造が実現できる可能性が高いことがあげられる。ハッシュ関数が 80 ビット出力と非常に短いため、 Birthday Attack により 2^{40} オーダーの計算でハッシュ値の衝突を引き起こすことができ、それを利用して適応的選択メッセージ攻撃による存在的偽造を実現できる可能性が高い。詳細な手順を 2.2節に述べる。自己評価書には、適応的選択メッセージ攻撃による存在的偽造不可であることが証明されているが、これはハッシュ関数をランダムオラクルと仮定しての結論であり、実際に用いられるハッシュ関数は衝突困難性を満たしているとは言いがたく、仮定と実際との乖離が大きいために不整合が起こっていると考えられる。本来の SHA-1 の出力値 160bit を 80 ビットに削ったのは、その分メッセージを回復し署名文全体の長さを短くするためであり、メッセージ回復のための設計方針が安全性を犠牲にしているといえる。なお、 MY-ELLTY ECMR-192-h はハッシュ値を 96 ビットとしており、この場合は 2^{48} オーダーの計算となるが、依然として適応的選択メッセージ攻撃による存在的偽造を実現できる可能性は高い。

仕様書の問題としては、仕様の不備、記述の誤り、記述もしくは考慮が不十分な点がいくつかあげられる。仕様の不備としては、仕様書で規定されていないと、異なる実装者間でインターオペラビリティを実現することができないと考えられる事項を列挙する。他にも記述の誤りおよび不十分と思われる点を列挙する。

2 安全性の問題

2.1 問題点

技術仕様書 13 頁には以下の記述がある。

「ハッシュ関数の代表例として、 SHA-1(Secure Hash Algorithm) を使用する。そして出力 160 ビットのうち必要なビット数 (80 ビット) を用いる。」

また、自己評価書 4 頁には以下の記述がある。

「メッセージ m を得るためには、 $h(m)$ から、 m を求める必要がある。ランダムに m を設定して、 $h(m)$ と一致するものを得るためには、ハッシュ関数が 80 ビットであるために、 2^{80} 回のオーダーの計算が必要となる。楕円曲線の離散対数問題の安全性は、 2^{80} のオーダーであるので、この攻撃の計算量は、楕円曲線の離散対数問題を解く計算量と同等であることになる。」

この記述は、与えられたハッシュ値を持つようなメッセージを探すことによる署名偽造攻撃は考慮しているが、同じハッシュ値を持つような 2 つのメッセージを探すことによる署名偽造攻撃 (Birthday Attack もしくは collision attack) は考慮していない。 Birthday Paradox に基づく Birthday Attack の計算量は、 $\sqrt{\frac{2^{80}\pi}{2}} \approx 1.25 \times 2^{40}$ 回のオーダーと見積もることができるため、安全性は遥かに低下する。

以下に攻撃手順詳細を述べる。

2.2 攻撃法詳細

1. 2^{40} 個のメッセージを生成する。ただし、本文 (最低限、先頭部 80 ビット) は同じものとし、最終部の通し番号のような部分 (例えば 12 桁の数字。 $2^{40} \approx 10^{12}$) を変えるようにする。
2. 各々のメッセージのハッシュ値を計算する。
3. ハッシュ値が一致する 2 つのメッセージ m_1, m_2 を得る。
4. m_1 の正しい署名 (r_1, s_1) を入手する。
5. (r_1, s_1) をメッセージ m_2 の署名として使用する。

署名仕様では、メッセージ回復のため、メッセージ先頭 80 ビットを署名プリミティブに渡す。手順 1においては、偽造のために、最低限先頭 80 ビットを一致させておく必要がある。また、この方法によって、偽造署名を作るメッセージを完全にではないが、ある程度の部分を攻撃者がコントロールすることができる。

手順 2では、 2^{40} オーダー回のハッシュ値の計算が必要であるが、計算機の並列度を上げることにより、短期間に達成可能な範囲である [1]。

手順 3は、計算したハッシュ値を保存するための記憶領域が問題となるが、 1 エントリにつき、順序付け番号 (40 ビット)、通し番号 (40 ビット)、ハッシュ値 (80 ビット) の計 20 バイト必要とすると、約 20×2^{40}B$\approx$ 20TB となり、実現可能な値である。また、衝突するハッシュ値を見つけるための操作は、ソーティングアルゴリズムにより、 $t = 2^{40}$ をエントリ数とすると、 $O(t \log t)$ 回の比較となり、これも実現可能な範囲である。

手順 4は、適応的選択メッセージ攻撃を許すという仮定において可能となる。

技術仕様書 7 頁では、メッセージ m の先頭 80 ビットを m_{re} で表し、 m のハッシュ値を h で表したとき、 $d = m_{re} \parallel h$ を署名生成プリミティブに渡すことにより、署名 (r, s) を計算するとある。この仕様より、署名対象メッセージの先頭 80 ビットおよびメッセージ全体のハッシュ値が一致していれば、同じ署名を得ることは明らかである。

2.3 安全性の考察

前節の結果より、本署名スキームは適応的選択メッセージ攻撃による存在的偽造が可能と思われる一方、自己評価書においは Forking Lemma を用いて、適応的選択文書攻撃に対して存在的偽造不

可であることが証明されている。この不整合は、証明においてはハッシュ関数をランダムオラクルモデルと仮定しているにもかかわらず、仕様においてはハッシュ値 80 ビットという、衝突困難性を満たすとは言いがたいハッシュ関数を使用しており、その乖離が大きいためである。

本署名スキームでは、メッセージ回復機能のために、ハッシュ値を 80 ビット削り、そこをメッセージ 80 ビットで置き換えている。この回復されるメッセージ 80 ビット分、署名文を短くすることができるというのが、提案者の訴求点である。見方を変えれば、ハッシュ関数の衝突困難性を犠牲にして、署名文を短くしているといえる。すなわち、本メッセージ回復署名の設計方針・訴求点が安全性を犠牲にしているととらえることができる。

なお、 MY-ELLTY ECMR-192-h はハッシュ値を 96 ビットとしており、この場合は 2^{48} オーダーの計算となるが、依然として適応的選択メッセージ攻撃による存在的偽造を実現できる可能性は高い。

NIST(National Institute of Standards and Technology) は、 SHA-1 の 160 ビット出力では collision attack に対して 80 ビット相当のセキュリティしか提供できないとして、 AES の鍵サイズ 128、192、 256 に相当するレベルのハッシュ関数として、 SHA-256、 SHA-384、 SHA-512 を FIPS として提案する動きにある [2]。これらと比較しても、ハッシュ値 80 ビットは、共通鍵サイズ 40 ビット相当のセキュリティであり、短かすぎて安全とはいえない ([3]165 頁、 430 頁)。

3 仕様書の問題

3.1 仕様の不備

以下の事項は、署名者と検証者の間で予め合意しておく事項とされていたり (下記 1)、記述がない (下記 2) ものであるが、これらは実装時に必要な情報である。これらの事項が仕様書で規定されていなければ、異なる実装者間でインターオペラビリティを実現することは困難である。

1. 技術仕様 7 頁。 80 ビット未満の長さのメッセージの 80 ビット長への拡張方法。これが規定されなければ、メッセージ m が 80 ビット未満のとき、 m_{re} を一意に定めることができない。

2. 技術仕様 13 頁。「(SHA-1 の) 出力 160 ビットのうち必要なビット数 (80 ビット) を用いる。」とあるが、どの 80 ビットを用いるのか。これが規定されなければ、ハッシュ値 h を一意に定めることができない。

3.2 記述の誤り

以下に、記述の誤りをあげる。最も中核となる署名プリミティブ仕様部にも誤りが 2 箇所ある (下記 2、 3)。これらの誤りが仕様検討不足によるものか、単なる記述ミスかの判断はつかないが、正誤表には記載されていない。

1. 技術仕様書 3 頁、 10 頁「c は p のビットサイズの半分より小さい」
   =>「c のビットサイズは、 p のビットサイズの半分より小さい」

2. 技術仕様書 8 頁「$0 < k < q$」
   =>「$1 < k < q$」
   $k = 1$ ならば、 $s = \frac{-1}{x+1} (\mathrm{mod}\ q)$ となる。検証時に、 $\frac{1+r'+s}{r'}G + \frac{s}{r'}Y = kG$ であるので、この点が G と一致するときに、 $x = -\frac{1}{s} - 1 (\mathrm{mod}\ q)$ を計算することにより、秘密鍵 x が露呈する。

3. 技術仕様書 9 頁「$0 < x < q$」
   =>「$0 < x < q - 1$」

$x = q - 1$ ならば、 $s = \frac{r'k - r' - 1}{x+1} (\mathrm{mod}\ q)$ の計算において零除算となり、署名生成処理が失敗する。

4. 技術仕様書 12 頁「t_2」
   =>「t_3」

3.3 記述もしくは考慮が不十分

1. 技術仕様書 3 頁、 10 頁、自己評価書 2 頁
   位数 q が満たすべき条件について、技術仕様書 4 頁に「なお、本技術仕様書では位数が素数となるものを利用する。」とあるだけで、その根拠が示されていない。自己評価書 2 頁には、 Pohlig-Hellman 攻撃について言及されているが、その節では Parallel Pollard ρ 法の計算量が $\sqrt{\frac{\pi q}{2}}$ であると述べられているだけであり、 q が素数でなければならないことに関して言及がない。 Pohlig-Hellman 攻撃に関する考慮の記述が不十分である。

2. 技術仕様書 3.6 節で述べられている高速化アルゴリズムの実装を必須にしているような記述が見受けられる。例えば、技術仕様書 10 頁には、
   「ここでは定義体の演算を実装するために用いるアルゴリズムについて説明する」
   11 頁には、
   「内部での処理は逆元演算が不要である projective 座標系を採用する」
   12 頁には、
   「楕円スカラ倍の演算には 2 種類の演算を用いる」
   などとある。

   これらの技法は、知的財産権などを考慮すると仕様で実装を義務づけるべきものではないと考えられる。これらの技術を実装時に使用するか否かは実装者が選択することができ、仕様としてこれらの技術の実装を必須としているのではないことを明記すべきである。もしくは、これらの技術の実装を必須としているのなら、その論拠を明記すべきである。

3. 技術仕様書 3.6.1 節で示された楕円曲線パラメータの選択方法が記されていない。具体的に与えられている楕円曲線パラメータはこの 1 つだけであるため、落とし戸の懸念をユーザに与えかねない。

4. 自己評価書 3 頁、 2.2.2 節において、有限体上の離散対数問題に帰着するときの提案パラメータにおける k の値を求める方法・計算が記述されていない。

4 まとめ

MY-ELLTY ECMR-160-h、 MY-ELLTY ECMR-192-h、 MY-ELLTY ECMR-OEF-h の安全性に関する評価を行なった。ハッシュ値出力が 80 ビットと非常に短いため、 Birthday Attack により、 2^{40} オーダーの計算で、適応的選択メッセージ攻撃による存在的署名偽造が実現できる可能性が高いことを指摘した。本署名スキームでは、メッセージ回復機能のために、ハッシュ値を 80 ビット削り、そこをメッセージ 80 ビットで置き換えている。ハッシュ関数の衝突困難性を犠牲にして、署名文を短くしているということであり、本メッセージ回復署名の設計方針・訴求点が安全性を犠牲にしていると考えられる。自己評価書では、適応的選択メッセージ攻撃による存在的偽造不可が証明されているが、これはハッシュ関数がランダムオラクルと仮定しての結論であり、実際に使用するハッシュ関数は衝突困難性を満たしているとは言いがたく、不整合が起きている。なお、 MY-ELLTY ECMR-192-h はハッシュ値を 96 ビットとしており、この場合は 2^{48} オーダーの計算となるが、依然として適応的選択メッセージ攻撃による存在的偽造を実現できる可能性は高い。

その他、インターオペラビリティを実現できない仕様の不備、記述の誤り、考慮不足なども散見された。

以上

References

[1] A. Bosselaers, Even faster hashing on the Pentium, Katholieke Universiteit Leuven, manuscript, May 13, 1997. http://www.esat.kuleuven.ac.be/~ bosselae/fast.html

[2] http://csrc.nist.gov/cryptval/shs.html

[3] B. Schneier, Applied Cryptography second edition, John Wiley & Sons, Inc. 1996