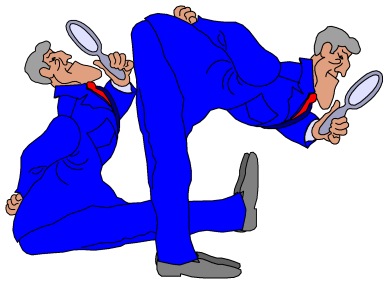
9. Autentičnost a digitální podpisy

Jan Paseka

Ústav matematiky a statistiky
Masarykova univerzita

6. prosince 2021

O čem to bude



Úvod I

FI V předchozích kapitolách jsme předvedli metody, které mohou pomoci proti **pasívnímu útoku**; pomocí zašifrování lze data pro nepovolané osoby udělat nečitelná. Téma této kapitoly je věnováno metodám proti **aktivnímu útoku**.

Úvod I

FI V předchozích kapitolách jsme předvedli metody, které mohou pomoci proti **pasívnímu útoku**; pomocí zašifrování lze data pro nepovolané osoby udělat nečitelná. Téma této kapitoly je věnováno metodám proti **aktivnímu útoku**.

Kryptografický protokol specifikuje, jakým způsobem každá strana začíná a odpovídá na zprávy a to včetně chybných nebo ilegálních zpráv.

Úvod I

FI V předchozích kapitolách jsme předvedli metody, které mohou pomoci proti **pasívnímu útoku**; pomocí zašifrování lze data pro nepovolané osoby udělat nečitelná. Téma této kapitoly je věnováno metodám proti **aktivnímu útoku**.

Kryptografický protokol specifikuje, jakým způsobem každá strana začíná a odpovídá na zprávy a to včetně chybných nebo ilegálních zpráv.

Protokol může rovněž specifikovat požadavky na nastavení jako je např. nastavení knihovny veřejných klíčů. Strana, která se řídí protokolem, bude ochráněna proti jistým specifikovaným nebezpečím i v tom případě, že ostatní strany se protokolem neřídí.

Úvod II

Je zřejné, že Mr. X může způsobit podstatně větší škodu, jestliže neumí pouze pasívně číst data, nýbrž je dokonce aktivně změnit.

Úvod II

Je zřejné, že Mr. X může způsobit podstatně větší škodu, jestliže neumí pouze pasívně číst data, nýbrž je dokonce aktivně změnit.

Skutečně se u většiny dnešních aplikací v kryptologii požaduje autentičnost dat a ne jejich utajení.

Úvod II

Je zřejné, že Mr. X může způsobit podstatně větší škodu, jestliže neumí pouze pasívně číst data, nýbrž je dokonce aktivně změnit.

Skutečně se u většiny dnešních aplikací v kryptologii požaduje autentičnost dat a ne jejich utajení.

Je zvykem rozlišovat 3 základní typy problémů, které vzniknou z různých variant aktivního útoku. Odpovídající "***bezpečnostní architektura***" byla vytvořena institucí ***International Standards Organisation (ISO)*** v "Security Addendum k referenčnímu modelu ISO".

Úvod II

Je zřejné, že Mr. X může způsobit podstatně větší škodu, jestliže neumí pouze pasívně číst data, nýbrž je dokonce aktivně změnit.

Skutečně se u většiny dnešních aplikací v kryptologii požaduje autentičnost dat a ne jejich utajení.

Je zvykem rozlišovat 3 základní typy problémů, které vzniknou z různých variant aktivního útoku. Odpovídající "***bezpečnostní architektura***" byla vytvořena institucí ***International Standards Organisation (ISO)*** v "Security Addendum k referenčnímu modelu ISO".

Nejdříve je třeba ptát se, zda byla zpráva přenesena bez změny nebo zfalšování; jedná se o požadavek **integrity zprávy**.

Úvod III

Je-li Mr. X v pozici, že může měnit zprávy, nezbývá než doufat, že příjemce ***zpozoruje*** případnou změnu. Musí být schopen rozhodnout, zda byla zpráva změněna či nikoliv.

Úvod III

Je-li Mr. X v pozici, že může měnit zprávy, nezbývá než doufat, že příjemce ***zpozoruje*** případnou změnu. Musí být schopen rozhodnout, zda byla zpráva změněna či nikoliv.

Druhý typ útoku je prvnímu podobný; zde je položen důraz na otázku, zda si příjemce může být jistý, že zpráva skutečně pochází od údajného odesilatele. Mluvíme pak o **autentičnosti zprávy**.

Úvod III

Je-li Mr. X v pozici, že může měnit zprávy, nezbývá než doufat, že příjemce ***zpozoruje*** případnou změnu. Musí být schopen rozhodnout, zda byla zpráva změněna či nikoliv.

Druhý typ útoku je prvnímu podobný; zde je položen důraz na otázku, zda si příjemce může být jistý, že zpráva skutečně pochází od údajného odesilatele. Mluvíme pak o **autentičnosti zprávy**.

Poslední varianta je **autentičnost uživatele**: Může osoba dokázat svoji identitu? Příjemce potřebuje prostředek, aby se mohl přesvědčit o tom, že skutečně komunikuje s tou osobou, o které si myslí, že je s ní spojen.

Úvod II

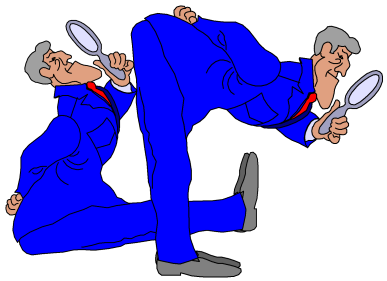
Po pěti letech byla šifra AES schválena jako nejvhodnější z patnácti návrhů. Dne 26. května 2002 začala být ke svému účelu používána jako federální standard USA. AES je první šifra, dostupná široké veřejnosti, která je zároveň uznána Národní bezpečností agenturou NSA.

Úvod II

Po pěti letech byla šifra AES schválena jako nejvhodnější z patnácti návrhů. Dne 26. května 2002 začala být ke svému účelu používána jako federální standard USA. AES je první šifra, dostupná široké veřejnosti, která je zároveň uznána Národní bezpečností agenturou NSA.

V současnoti se využívá k šifrování elektronické pošty, elektronického bankovnictví, různých druhů dálkové autentizace, čipových karet, elektronických peněz, přenosu hovorů v síti GSM, signálu wi-fi, bluetooth a satelitů.

O čem to bude



Symetrická autenticita I

Ochrana autentičnosti informace sestává z dvou následujících aspektů:

- ochrana původce informace neboli dle terminologie ISO autenticita původu dat,

Symetrická autenticita I

Ochrana autentičnosti informace sestává z dvou následujících aspektů:

- ochrana původce informace neboli dle terminologie ISO autenticita původu dat,
- skutečnost, že informace nebyla změněna neboli dle terminologie ISO integrita informace.

Symetrická autenticita I

Ochrana autentičnosti informace sestává z dvou následujících aspektů:

- ochrana původce informace neboli dle terminologie ISO autenticita původu dat,
- skutečnost, že informace nebyla změněna neboli dle terminologie ISO integrita informace.

První aspekt lze prezentovat tak, že je informace načítána např. z harddisku osobního počítače a my implicitně důvěřujeme zdroji informace.

Symetrická autenticita I

Ochrana autentičnosti informace sestává z dvou následujících aspektů:

- ochrana původce informace neboli dle terminologie ISO autenticita původu dat,
- skutečnost, že informace nebyla změněna neboli dle terminologie ISO integrita informace.

První aspekt lze prezentovat tak, že je informace načítána např. z harddisku osobního počítače a my implicitně důvěřujeme zdroji informace.

Jiným aspektem je časování, umístění do fronty vzhledem k jiným zprávám a určení zprávy.

Symetrická autenticita I

Ochrana autentičnosti informace sestává z dvou následujících aspektů:

- ochrana původce informace neboli dle terminologie ISO autenticita původu dat,
- skutečnost, že informace nebyla změněna neboli dle terminologie ISO integrita informace.

První aspekt lze prezentovat tak, že je informace načítána např. z harddisku osobního počítače a my implicitně důvěřujeme zdroji informace.

Jiným aspektem je časování, umístění do fronty vzhledem k jiným zprávám a určení zprávy.

Až donedávna se obecně předpokládalo, že zašifrování informace je dostatečné k tomu, aby se prokázala její autenticita.

Symetrická autenticita II

Použitý argument byl ten, že pokud šifrový text dal po dešifrování smysluplnou informaci, tato by měla vzniknout od někoho, kdo zná tajný klíč, což garantuje autenticitu zprávy a odesílatele.

Symetrická autenticita II

Použitý argument byl ten, že pokud šifrový text dal po dešifrování smysluplnou informaci, tato by měla vzniknout od někoho, kdo zná tajný klíč, což garantuje autenticitu zprávy a odesílatele.

Ve dvou příkladech ukážeme, že tato víra není správná: ochrana integrity závisí na šifrovacím algoritmu a na módu, ve kterém je algoritmus použit.

Symetrická autenticita II

Použitý argument byl ten, že pokud šifrový text dal po dešifrování smysluplnou informaci, tato by měla vzniknout od někoho, kdo zná tajný klíč, což garantuje autenticitu zprávy a odesílatele.

Ve dvou příkladech ukážeme, že tato víra není správná: ochrana integrity závisí na šifrovacím algoritmu a na módu, ve kterém je algoritmus použit.

Vernamova šifra, kde je náhodný klíč přičítán modulo 2 k šifrovému textu nám poskytuje perfektní bezpečnost, ale aktivní útočník může změnit libovolný bit zdrojového textu tím, že změní odpovídající bit šifrového textu.

Symetrická autenticita II

Použitý argument byl ten, že pokud šifrový text dal po dešifrování smysluplnou informaci, tato by měla vzniknout od někoho, kdo zná tajný klíč, což garantuje autenticitu zprávy a odesílatele.

Ve dvou příkladech ukážeme, že tato víra není správná: ochrana integrity závisí na šifrovacím algoritmu a na módu, ve kterém je algoritmus použit.

Vernamova šifra, kde je náhodný klíč přičítán modulo 2 k šifrovému textu nám poskytuje perfektní bezpečnost, ale aktivní útočník může změnit libovolný bit zdrojového textu tím, že změní odpovídající bit šifrového textu.

Tato informace analogicky platí pro libovolnou přičítací proudovou šifru a pro OFB mód (Output FeedBack) každé blokové šifry.

Symetrická autenticita III

Částečně toto platí i pro případ, že šifra je použita CFB módu (Cipher FeedBack) nebo CBC módu (Cipher Block Chaining).

Symetrická autenticita III

Částečně toto platí i pro případ, že šifra je použita CFB módu (Cipher FeedBack) nebo CBC módu (Cipher Block Chaining).

Je-li zdrojový text delší než jeden blok zašifrován pomocí blokové šifry v ECB módu, aktivní útočník může snadno přeuspořádat bloky.

Symetrická autenticita III

Částečně toto platí i pro případ, že šifra je použita CFB módu (Cipher FeedBack) nebo CBC módu (Cipher Block Chaining).

Je-li zdrojový text delší než jeden blok zašifrován pomocí blokové šifry v ECB módu, aktivní útočník může snadno přeuspořádat bloky.

Jiným příkladem zranitelnosti aktivním útočníkem je zdrojový text zašifrovaný pomocí CFB módu. Vzhledem k synchronizačním vlastnostem každá modifikace šifrového textu způsobí odpovídající modifikaci zdrojového textu a následně zkomolí následující části zdrojového textu. Poté co chyba opustí FB registr, bude šifrový text opět správně dešifrován.

Symetrická autenticita III

Částečně toto platí i pro případ, že šifra je použita CFB módu (Cipher FeedBack) nebo CBC módu (Cipher Block Chaining).

Je-li zdrojový text delší než jeden blok zašifrován pomocí blokové šifry v ECB módu, aktivní útočník může snadno přeuspořádat bloky.

Jiným příkladem zranitelnosti aktivním útočníkem je zdrojový text zašifrovaný pomocí CFB módu. Vzhledem k synchronizačním vlastnostem každá modifikace šifrového textu způsobí odpovídající modifikaci zdrojového textu a následně zkomolí následující části zdrojového textu. Poté co chyba opustí FB registr, bude šifrový text opět správně dešifrován.

Je-li ale modifikována poslední část šifrového textu, je zcela nemožné najít tuto modifikaci. Pokud se zkomolení vyskytne uprostřed zdrojového textu, lze chybu detekovat pomocí redundance.

Symetrická autenticita IV

V jiných módech (jako např. CBC mód) je každý šifrový text složitou funkcí předchozích bitů zdrojového textu a nějaké počáteční hodnoty.

Symetrická autenticita IV

V jiných módech (jako např. CBC mód) je každý šifrový text složitou funkcí předchozích bitů zdrojového textu a nějaké počáteční hodnoty.

Pokud modifikace jednoho bitu šifrového textu způsobí zkomolení t bitů zdrojového textu, pravděpodobnost, že že nový zdrojový text bude akceptován jako smysluplný, je rovna 2^{-tD}, kde D je redundance informace.

Symetrická autenticita IV

V jiných módech (jako např. CBC mód) je každý šifrový text složitou funkcí předchozích bitů zdrojového textu a nějaké počáteční hodnoty.

Pokud modifikace jednoho bitu šifrového textu způsobí zkomolení t bitů zdrojového textu, pravděpodobnost, že že nový zdrojový text bude akceptován jako smysluplný, je rovna 2^{-tD}, kde D je redundance informace.

V případě přirozeného jazyka zakódovaného pomocí 5 bitů na charakter je redundance na bit $D \simeq 0.74$ a tato pravděpodobnost je rovna $2^{-22.2}$ pro $t = 30$.

Symetrická autenticita IV

V jiných módech (jako např. CBC mód) je každý šifrový text složitou funkcí předchozích bitů zdrojového textu a nějaké počáteční hodnoty.

Pokud modifikace jednoho bitu šifrového textu způsobí zkomolení t bitů zdrojového textu, pravděpodobnost, že že nový zdrojový text bude akceptován jako smysluplný, je rovna 2^{-tD}, kde D je redundance informace.

V případě přirozeného jazyka zakódovaného pomocí 5 bitů na charakter je redundance na bit $D \simeq 0.74$ a tato pravděpodobnost je rovna $2^{-22.2}$ pro $t = 30$.

Avšak, je-li $D = 0$ a zašifrování neposkytuje žádnou autenticitu, jsou všechny zprávy smysluplné a to nezávisle na šifrovacím algoritmu nebo na módu šifrování.

Symetrická autenticita V

To pak znamená, že útočník může modifikovat zprávy nebo padělat zprávy dle svého výběru.

Symetrická autenticita V

To pak znamená, že útočník může modifikovat zprávy nebo padělat zprávy dle svého výběru.

Omezení je pak to, že útočník neví dopředu, co bude obsahem odpovídajícího zdrojového textu, ale pro mnohé aplikace lze takovýto útok považovat za zdroj vážných problémů.

Symetrická autenticita V

To pak znamená, že útočník může modifikovat zprávy nebo padělat zprávy dle svého výběru.

Omezení je pak to, že útočník neví dopředu, co bude obsahem odpovídajícího zdrojového textu, ale pro mnohé aplikace lze takovýto útok považovat za zdroj vážných problémů.

Poznamenejme, že i v případě existence redundance se požaduje kontrola lidským faktorem nebo vhodným počítačovým programem.

Symetrická autenticita V

To pak znamená, že útočník může modifikovat zprávy nebo padělat zprávy dle svého výběru.

Omezení je pak to, že útočník neví dopředu, co bude obsahem odpovídajícího zdrojového textu, ale pro mnohé aplikace lze takovýto útok považovat za zdroj vážných problémů.

Poznamenejme, že i v případě existence redundance se požaduje kontrola lidským faktorem nebo vhodným počítačovým programem.

Abychom zajistili integritu zprávy, je nutno přidat speciální redundanci, a je-li informace spojena s původcem zprávy, musí být použit v tomto procesu tajný klíč (to předpokládá spojení osoby a jejího klíče) nebo zvláštního kanálu pro zajištění integrity.

Symetrická autenticita VI

Můžeme pak identifikovat dvě základní metody.

Symetrická autenticita VI

Můžeme pak identifikovat dvě základní metody.

- První metoda je analogická metodě symetrické šifry, kde utajení velkého množství dat je založeno na utajení a autenticitě krátkého klíče. V tomto případě autenticita informace závisí na utajení a autenticitě klíče.

Symetrická autenticita VI

Můžeme pak identifikovat dvě základní metody.

- První metoda je analogická metodě symetrické šifry, kde utajení velkého množství dat je založeno na utajení a autenticitě krátkého klíče. V tomto případě autenticita informace závisí na utajení a autenticitě klíče.

  Abychom dosáhli tohoto účelu, informace se zkomprimuje na kvantitu pevné délky, kterou nazýváme ***hešovacím kódem***. Poté se hešovací kód připojí k informaci. Funkce, která provede tuto operaci komprese, se nazývá ***hešovací funkce***.

Symetrická autenticita VI

Můžeme pak identifikovat dvě základní metody.

- První metoda je analogická metodě symetrické šifry, kde utajení velkého množství dat je založeno na utajení a autenticitě krátkého klíče. V tomto případě autenticita informace závisí na utajení a autenticitě klíče.

  Abychom dosáhli tohoto účelu, informace se zkomprimuje na kvantitu pevné délky, kterou nazýváme ***hešovacím kódem***. Poté se hešovací kód připojí k informaci. Funkce, která provede tuto operaci komprese, se nazývá ***hešovací funkce***.

  Základní myšlenkou zabezpečení integrity je ***přidat redundanci*** k informaci. Přítomnost redundance dovoluje příjemci provést rozlišení autentické informace a podvodné informace.

Symetrická autenticita VII

Abychom garantovali původ dat, je nutno v procesu použít tajný klíč. Tajný klíč může být obsažen v procesu komprese nebo může být použit, aby ochránil hešovací kód a/nebo informaci. V prvním případě mluvíme o MACu (***Message Authentication Code***), zatímco v druhém případě se hešovací kód nazývá MDC (***Manipulation Detection Code***).

Symetrická autenticita VII

Abychom garantovali původ dat, je nutno v procesu použít tajný klíč. Tajný klíč může být obsažen v procesu komprese nebo může být použit, aby ochránil hešovací kód a/nebo informaci. V prvním případě mluvíme o MACu (***Message Authentication Code***), zatímco v druhém případě se hešovací kód nazývá MDC (***Manipulation Detection Code***).

- Druhá metoda sestává na zajištění autenticity (jak integrity a autenticity původu) informace o autenticitě MDC. Typickým příkladem této metody je uživatel počítače, který počítá MDC pro všechny své důležité soubory. Může si pak uložit soubor všech MDC na disketě, kterou si bezpečně uschová. Pokud tyto soubory zašle vzdálenému příteli, může jednoduše poslat soubory a sdělit příteli po telefonu jejich MDC. Autenticita telefonního kanálu je zajištěna hlasovou identifikací.

Symetrická autenticita VIII

Přidání redundance není jistě dostatečné. Speciální důraz musíme klást na obranu proti útokům na vysoké úrovni, jako je například opakování autentifikované zprávy.
Oba případy nefungují, pokud si odesílatel a příjemce navzájem nedůvěřují. V prvním případě sdílejí stejný tajný klíč. Pokud jedna ze stran tvrdí, že informace byla změněna druhou stranou, nemůže soudce rozhodnout, kdo má pravdu, i když obě strany vydají společný tajný klíč. Druhý přístup může pouze zajistit nepřevzetí, pokud obě strany věří autenticitě MDC: v praxi je to však obtížné realizovat, protože obě strany mají podobný přístup ke kanálu.

Asymetrická autenticita I

Jestliže chceme být ochráněni proti vnitřnímu napadnutí, potřebujeme elektronický ekvivalent podpisu. V tomto případě třetí strana bude schopna rozlišit dvě strany a to na základě skutečnosti, že způsobilosti obou stran jsou různé.

Asymetrická autenticita I

Jestliže chceme být ochráněni proti vnitřnímu napadnutí, potřebujeme elektronický ekvivalent podpisu. V tomto případě třetí strana bude schopna rozlišit dvě strany a to na základě skutečnosti, že způsobilosti obou stran jsou různé.

Pojem digitálního podpisu byl zaveden W. Diffiem a M. Hellmanem.

Asymetrická autenticita I

Jestliže chceme být ochráněni proti vnitřnímu napadnutí, potřebujeme elektronický ekvivalent podpisu. V tomto případě třetí strana bude schopna rozlišit dvě strany a to na základě skutečnosti, že způsobilosti obou stran jsou různé.

Pojem digitálního podpisu byl zaveden W. Diffiem a M. Hellmanem.

Požadavky na elektronický podpis jsou, že ***podpis závisí na podepisované informaci*** (protože není fyzicky spjat s dokumentem) a že ***podepsaný je jediná osoba, která je schopna vytvořit podpis*** (to znamená, že nikdo jiný nemůže zfalšovat podpis tj. podepsaný nemůže zapřít, že informaci podepsal právě on).

Asymetrická autenticita II

Digitální podpisové schéma sestává z následujících prvků:

- inicializační fáze (např. generování klíče a obecné nastavení),
- procesu podpisu, kdy je vytvořen podpis,
- procesu verifikace, kdy příjemce (nebo soudce) ověří, zda je podpis správný.

Asymetrická autenticita II

Digitální podpisové schéma sestává z následujících prvků:

- inicializační fáze (např. generování klíče a obecné nastavení),
- procesu podpisu, kdy je vytvořen podpis,
- procesu verifikace, kdy příjemce (nebo soudce) ověří, zda je podpis správný.

Digitální podpis v tomto smyslu lze vytvořit pomocí zařízení bezpečných proti falšování, konvenčních jednosměrných funkcí nebo technik veřejného klíče.

Asymetrická autenticita II

Digitální podpisové schéma sestává z následujících prvků:

- inicializační fáze (např. generování klíče a obecné nastavení),
- procesu podpisu, kdy je vytvořen podpis,
- procesu verifikace, kdy příjemce (nebo soudce) ověří, zda je podpis správný.

Digitální podpis v tomto smyslu lze vytvořit pomocí zařízení bezpečných proti falšování, konvenčních jednosměrných funkcí nebo technik veřejného klíče.

Poznamenejme dále, že bylo definováno několik zobecnění – např. s různými stupni bezpečnosti a více hráči ve hře.

Asymetrická autenticita III

Příklady takovýchto jsou následující: ***libovolné podpisy***, kde proces podpis a verifikace zahrnuje interakci s třetí stranou,

Asymetrická autenticita III

Příklady takovýchto jsou následující: ***libovolné podpisy***, kde proces podpis a verifikace zahrnuje interakci s třetí stranou, ***skupinové podpisy***, kde podpisující a/nebo kontroloři jsou členy skupiny,

Asymetrická autenticita III

Příklady takovýchto jsou následující: ***libovolné podpisy***, kde proces podpis a verifikace zahrnuje interakci s třetí stranou, ***skupinové podpisy***, kde podpisující a/nebo kontroloři jsou členy skupiny, ***slepé podpisy***, kde podpisující podepíše "slepou " nebo " maskovanou" zprávu a

Asymetrická autenticita III

Příklady takovýchto jsou následující: ***libovolné podpisy***, kde proces podpis a verifikace zahrnuje interakci s třetí stranou, ***skupinové podpisy***, kde podpisující a/nebo kontroloři jsou členy skupiny, ***slepé podpisy***, kde podpisující podepíše "slepou " nebo " maskovanou" zprávu a ***neviditelné*** nebo ***nepopiratelné*** zprávy, kde lze podpis verifikovat pouze ve spolupráci s podpisujícím.

Message-Authentication-Code I

Připomeňme si, že při integritě a autentičnosti zprávy jde o to, abychom vyvinuli metody, které příjemci umožní rozhodnout, zda zpráva došla neporušená a autentická. K tomu potřebuje příjemce něco, s čím může být zpráva ověřena: Potřebuje dodatečnou informaci od odesilatele.

Message-Authentication-Code I

Připomeňme si, že při integritě a autentičnosti zprávy jde o to, abychom vyvinuli metody, které příjemci umožní rozhodnout, zda zpráva došla neporušená a autentická. K tomu potřebuje příjemce něco, s čím může být zpráva ověřena: Potřebuje dodatečnou informaci od odesilatele.

Takový informační blok se nazývá **kryptografický zkušební součet, kryptografický otisk prvku** neboli **Message-Authentication-Code**, zkráceně **MAC**.

Message-Authentication-Code I

Připomeňme si, že při integritě a autentičnosti zprávy jde o to, abychom vyvinuli metody, které příjemci umožní rozhodnout, zda zpráva došla neporušená a autentická. K tomu potřebuje příjemce něco, s čím může být zpráva ověřena: Potřebuje dodatečnou informaci od odesilatele.

Takový informační blok se nazývá **kryptografický zkušební součet, kryptografický otisk prvku** neboli **Message-Authentication-Code**, zkráceně **MAC**.

Protokol k vytvoření a verifikaci kryptografického zkušebního součtu je založen na použití tajného klíče k, který je znám jak odesilateli tak příjemci, a kryptografickém algoritmu A, který budeme v dalším diskutovat.

Message-Authentication-Code II

Odesílatel neposílá pouze holou zprávu M, nýbrž dodatečně příslušný MAC; ten se vypočte pomocí klíče k algoritmem A ze zprávy M následovně:

$$\text{MAC} = A_k(M).$$

Message-Authentication-Code II

Odesílatel neposílá pouze holou zprávu M, nýbrž dodatečně příslušný MAC; ten se vypočte pomocí klíče k algoritmem A ze zprávy M následovně:

$$\text{MAC} = A_k(M).$$

Poznamenejme, že M je odesíláno nezašifrované, protože cílem odesílatele není utajit obsah zprávy, nýbrž zprávu zabezpečit. Pokud chceme navíc důvěrnost, musí být m a MAC zašifrovány.

Message-Authentication-Code II

Odesílatel neposílá pouze holou zprávu M, nýbrž dodatečně příslušný MAC; ten se vypočte pomocí klíče k algoritmem A ze zprávy M následovně:

$$\text{MAC} = A_k(M).$$

Poznamenejme, že M je odesíláno nezašifrované, protože cílem odesílatele není utajit obsah zprávy, nýbrž zprávu zabezpečit. Pokud chceme navíc důvěrnost, musí být m a MAC zašifrovány.

Nyní přijde na řadu příjemce. Jeho zájem je zjistit, zda přijatá zpráva souhlasí se zprávou odeslanou a zda skutečně pochází od uvedeného odesilatele.

Message-Authentication-Code III

Aby to provedl, simuluje proceduru odesilatele: Použije algoritmus A s klíčem k na přijatou zprávu M' a prověří, zda výsledek souhlasí s obrženým MACem.

Message-Authentication-Code III

Aby to provedl, simuluje proceduru odesilatele: Použije algoritmus A s klíčem k na přijatou zprávu M' a prověří, zda výsledek souhlasí s obrženým MACem.

Je-li $A_k(M') \neq MAC'$, ví příjemce, že se "***něco***" stalo: proto neakceptuje zprávu jako autentickou a odmítne ji.

Message-Authentication-Code III

Aby to provedl, simuluje proceduru odesilatele: Použije algoritmus A s klíčem k na přijatou zprávu M' a prověří, zda výsledek souhlasí s obrženým MACem.

Je-li $A_k(M') \neq MAC'$, ví příjemce, že se "***něco***" stalo: proto neakceptuje zprávu jako autentickou a odmítne ji.

Je-li ale $A_k(M') = MAC'$, může si být dostatečně jistý, že zpráva nebyla změněna. Přirozeně tato jistota závisí ve velké míře na kvalitě algoritmu A a velikosti množiny možných klíčů. Představy o MAC–mechanismu jsou následující:

- Podvodu ze strany Mr. X bude zamezeno, protože nezná klíč k. Musel by totiž spočítat odpovídající MAC pro svou zprávu.

Message-Authentication-Code IV

- Příjemce může pouze ***rozpoznat***, či je zpráva neporušená a autentická; v záporném případě nemá žádnou možnost zrekonstruovat původní zprávu. To znamená, že v tomto případě je nutný nový přenos zprávy.
- MAC–mechanismus je metoda k dosažení integrity a autentičnosti. Už jsme viděli, že lze poznat integritu. Pokud proběhne verifikace kladně, je příjemce rovněž přesvědčen o autentičnosti zprávy, protože odesílatel je jedinou jinou instancí, která zná tajný klíč.

Message-Authentication-Code V

Jaké algoritmy A můžeme použít k výpočtu MACu?

Message-Authentication-Code V

Jaké algoritmy A můžeme použít k výpočtu MACu?

Okamžitá odpověď je jednoduchá: použijme jednoduše šifrovací algoritmus, přičemž MAC je kryptogram, který odpovídá zprávě M.

Message-Authentication-Code V

Jaké algoritmy *A* můžeme použít k výpočtu MACu?

Okamžitá odpověď je jednoduchá: použijme jednoduše šifrovací algoritmus, přičemž MAC je kryptogram, který odpovídá zprávě *M*.

Odhlédneme-li od toho, že takovýto slabý algoritmus není vhodné doporučit, má tento návrh tu nevýhodu, že přenášená data jsou dvojnásobně delší než "vlastní zpráva".

Message-Authentication-Code V

Jaké algoritmy *A* můžeme použít k výpočtu MACu?

Okamžitá odpověď je jednoduchá: použijme jednoduše šifrovací algoritmus, přičemž MAC je kryptogram, který odpovídá zprávě *M*.

Odhlédneme-li od toho, že takovýto slabý algoritmus není vhodné doporučit, má tento návrh tu nevýhodu, že přenášená data jsou dvojnásobně delší než "vlastní zpráva".

Přirozeně každý MAC prodlouží zprávu, ale chtěli bychom délku tohoto dodatečného bloku držet v nějakých rozumných mezích.

Message-Authentication-Code V

Jaké algoritmy *A* můžeme použít k výpočtu MACu?

Okamžitá odpověď je jednoduchá: použijme jednoduše šifrovací algoritmus, přičemž MAC je kryptogram, který odpovídá zprávě *M*.

Odhlédneme-li od toho, že takovýto slabý algoritmus není vhodné doporučit, má tento návrh tu nevýhodu, že přenášená data jsou dvojnásobně delší než "vlastní zpráva".

Přirozeně každý MAC prodlouží zprávu, ale chtěli bychom délku tohoto dodatečného bloku držet v nějakých rozumných mezích.

V praxi používáme k výpočtu MACu také šifrovací algoritmus, ale ne přímo, nýbrž v tzv. **Cipher-Block-Chaining módu**.

Message-Authentication-Code VI

Představme si šifrovací algoritmus f (v praxi se většinou používá algoritmus DES nebo AES), který zobrazuje bloky zprávy složených z n znaků pomocí nějakého klíče K na bloky kryptogramu, rovněž složených z n znaků (typická hodnota je $n = 64$).

Message-Authentication-Code VI

Představme si šifrovací algoritmus f (v praxi se většinou používá algoritmus DES nebo AES), který zobrazuje bloky zprávy složených z n znaků pomocí nějakého klíče K na bloky kryptogramu, rovněž složených z n znaků (typická hodnota je $n = 64$).

Abychom mohli vypočítat MAC, rozdělíme zprávu M do bloků $M_1, M_2, \ldots, M_s$ délky n.

Message-Authentication-Code VI

Představme si šifrovací algoritmus f (v praxi se většinou používá algoritmus DES nebo AES), který zobrazuje bloky zprávy složených z n znaků pomocí nějakého klíče K na bloky kryptogramu, rovněž složených z n znaků (typická hodnota je $n = 64$).

Abychom mohli vypočítat MAC, rozdělíme zprávu M do bloků $M_1, M_2, \ldots, M_s$ délky n.

Pak aplikujeme f na blok M_1 a obdržíme první blok kryptogramu $C_1 = f_K(M_1)$.

Message-Authentication-Code VI

Představme si šifrovací algoritmus f (v praxi se většinou používá algoritmus DES nebo AES), který zobrazuje bloky zprávy složených z n znaků pomocí nějakého klíče K na bloky kryptogramu, rovněž složených z n znaků (typická hodnota je $n = 64$).

Abychom mohli vypočítat MAC, rozdělíme zprávu M do bloků $M_1, M_2, \ldots, M_s$ délky n.

Pak aplikujeme f na blok M_1 a obdržíme první blok kryptogramu $C_1 = f_K(M_1)$.

Potom přičteme C_1 k M_2 a položíme $C_2 = f_K(C_1 \oplus M_2)$.

Message-Authentication-Code VI

Představme si šifrovací algoritmus f (v praxi se většinou používá algoritmus DES nebo AES), který zobrazuje bloky zprávy složených z n znaků pomocí nějakého klíče K na bloky kryptogramu, rovněž složených z n znaků (typická hodnota je $n = 64$).

Abychom mohli vypočítat MAC, rozdělíme zprávu M do bloků $M_1, M_2, \ldots, M_s$ délky n.

Pak aplikujeme f na blok M_1 a obdržíme první blok kryptogramu $C_1 = f_K(M_1)$.

Potom přičteme C_1 k M_2 a položíme $C_2 = f_K(C_1 \oplus M_2)$.

Tento postup opakujeme až skončíme výstupem $C_s = f_K(C_{s-1} \oplus M_s)$, který vybereme za MAC.

Message-Authentication-Code VII

Takto vypočtený MAC má následující přednosti:

- MAC má pevnou délku n nezávislou na délce zprávy.

Message-Authentication-Code VII

Takto vypočtený MAC má následující přednosti:

- MAC má pevnou délku n nezávislou na délce zprávy.
- MAC závisí na všech blocích zprávy.

Message-Authentication-Code VII

Takto vypočtený MAC má následující přednosti:

- MAC má pevnou délku n nezávislou na délce zprávy.
- MAC závisí na všech blocích zprávy.

Protože všechny možné zprávy jsou zkomprimovány na MACy pevné délky, má mnoho zpráv tentýž MAC.

Message-Authentication-Code VII

Takto vypočtený MAC má následující přednosti:

- MAC má pevnou délku n nezávislou na délce zprávy.
- MAC závisí na všech blocích zprávy.

Protože všechny možné zprávy jsou zkomprimovány na MACy pevné délky, má mnoho zpráv tentýž MAC.

To nepředstavuje žádný problém pro příjemce, protože ten nemusí rekonstruovat původní zprávu z MACu.

Message-Authentication-Code VII

Takto vypočtený MAC má následující přednosti:

- MAC má pevnou délku n nezávislou na délce zprávy.
- MAC závisí na všech blocích zprávy.

Protože všechny možné zprávy jsou zkomprimovány na MACy pevné délky, má mnoho zpráv tentýž MAC.

To nepředstavuje žádný problém pro příjemce, protože ten nemusí rekonstruovat původní zprávu z MACu.

Ne všechny algoritmy jsou vhodné k tomu, aby byl Mr. X postaven před nepřekonatelné problémy.

Message-Authentication-Code VIII

Algoritmus pro výpočet MACu by měl mít následující vlastnosti.

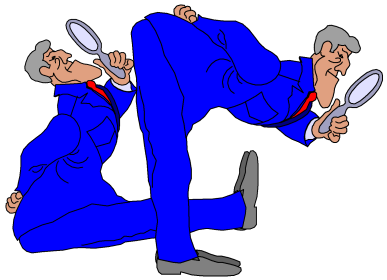
1. Mělo by být prakticky nemožné najít pro daný MAC odpovídající zprávu (pokud tato vlastnost platí, nazýváme algoritmus **jednosměrnou** (one-way) **funkcí**). "Prakticky nemožné" znamená, že s dnešními metodami a počítači by vyřešení problému trvalo příliš dloho (několik století).

Message-Authentication-Code VIII

Algoritmus pro výpočet MACu by měl mít následující vlastnosti.

1. Mělo by být prakticky nemožné najít pro daný MAC odpovídající zprávu (pokud tato vlastnost platí, nazýváme algoritmus **jednosměrnou** (one-way) **funkcí**). "Prakticky nemožné" znamená, že s dnešními metodami a počítači by vyřešení problému trvalo příliš dloho (několik století).
2. Mělo by být prakticky nemožné najít dvě zprávy, které mají tentýž MAC (jednosměrná funkce splňující tuto podmínku se nazývá **bezkolizní**).

O čem to bude



Definice jednosměrné funkce I

Je velmi obtížné podat precizní matematickou definici jednosměrné funkce. Neformálně je ***jednosměrná funkce*** funkce $f : S \to T$, kde S a T jsou množiny takové, že

(1) pro všechna $x \in S$ je $f(x)$ "***snadno***" vypočitatelné,

Definice jednosměrné funkce I

Je velmi obtížné podat precizní matematickou definici jednosměrné funkce. Neformálně je ***jednosměrná funkce*** funkce $f : S \to T$, kde S a T jsou množiny takové, že
(1) pro všechna $x \in S$ je $f(x)$ "***snadno***" vypočitatelné,
(2) máme-li k dispozici informaci, že $f(x) = y$, neexistuje žádný "***přiměřený***" způsob
jak získat (výpočtem) x pro "***dostatečně velké***" množství prvků y z T.

Definice jednosměrné funkce I

Je velmi obtížné podat precizní matematickou definici jednosměrné funkce. Neformálně je ***jednosměrná funkce*** funkce $f : S \to T$, kde S a T jsou množiny takové, že
(1) pro všechna $x \in S$ je $f(x)$ "***snadno***" vypočitatelné,
(2) máme-li k dispozici informaci, že $f(x) = y$, neexistuje žádný "***přiměřený***" způsob
jak získat (výpočtem) x pro "***dostatečně velké***" množství prvků y z T.

Pracovní slova zde jsou "***snadno***", "***přiměřeně***" a "***dostatečně velké***".

Definice jednosměrné funkce I

Je velmi obtížné podat precizní matematickou definici jednosměrné funkce. Neformálně je ***jednosměrná funkce*** funkce $f : S \to T$, kde S a T jsou množiny takové, že
(1) pro všechna $x \in S$ je $f(x)$ "***snadno***" vypočitatelné,
(2) máme-li k dispozici informaci, že $f(x) = y$, neexistuje žádný "***přiměřený***" způsob
jak získat (výpočtem) x pro "***dostatečně velké***" množství prvků y z T.

Pracovní slova zde jsou "***snadno***", "***přiměřeně***" a "***dostatečně velké***".

Je zřejmé, že je-li dáno $f(x)$, jeden způsob, jak získat x je prohledávat všechny možné hodnoty $x \in S$. Nepovažujeme to za přiměřené, protože S sestává obvykle z posloupnosti binárních řetězců délky $n \sim 200$.

Definice jednosměrné funkce II

Požadujeme, že výpočet pro nalezení x ze znalosti y je příliš dlouhotrvající nebo nákladný, kdykoliv y leží v "dosti velké" podmnožině množiny T.

Definice jednosměrné funkce II

Požadujeme, že výpočet pro nalezení x ze znalosti y je příliš dlouhotrvající nebo nákladný, kdykoliv y leží v "dosti velké" podmnožině množiny T.

Příklad. Elementárním příkladem kandidáta na jednosměrnou funkci je, pro dostatečně velké prvočíslo p, funkce $f(x)$, kde $f(x)$ je polynom nad tělesem $\mathbf{Z}_p$.

Definice jednosměrné funkce II

Požadujeme, že výpočet pro nalezení x ze znalosti y je příliš dlouhotrvající nebo nákladný, kdykoliv y leží v "dosti velké" podmnožině množiny T.

Příklad. Elementárním příkladem kandidáta na jednosměrnou funkci je, pro dostatečně velké prvočíslo p, funkce $f(x)$, kde $f(x)$ je polynom nad tělesem $\mathbf{Z}_p$.

Pak je relativně snadné vypočíst $f(x)$ $(1 \leq x \leq p-1)$, ale obvykle je těžké nalézt řešení rovnice

$$f(x) = y.$$

Definice jednosměrné funkce II

Požadujeme, že výpočet pro nalezení x ze znalosti y je příliš dlouhotrvající nebo nákladný, kdykoliv y leží v "dosti velké" podmnožině množiny T.

Příklad. Elementárním příkladem kandidáta na jednosměrnou funkci je, pro dostatečně velké prvočíslo p, funkce $f(x)$, kde $f(x)$ je polynom nad tělesem $\mathbf{Z}_p$.

Pak je relativně snadné vypočíst $f(x)$ $(1 \leq x \leq p-1)$, ale obvykle je těžké nalézt řešení rovnice

$$f(x) = y.$$

Výše uvedená nepřesná definice znamená, že to, co je jednosměrná funkce, se mění s dobou.

Definice jednosměrné funkce III

Například, výpočet požadující milión instrukcí a 10 000 slov paměti nemohl být v roce 1950 považován za snadný, ale nyní by trval několik sekund na osobním počítači.

Definice jednosměrné funkce III

Například, výpočet požadující milión instrukcí a 10 000 slov paměti nemohl být v roce 1950 považován za snadný, ale nyní by trval několik sekund na osobním počítači.

Tedy funkce považovaná v roce 1950 za jednosměrnou nemusí být za ni považovaná nyní.
Jedna metoda podání formální definice by mohlo být užitím fyzikálního přístupu.

Definice jednosměrné funkce III

Například, výpočet požadující milión instrukcí a 10 000 slov paměti nemohl být v roce 1950 považován za snadný, ale nyní by trval několik sekund na osobním počítači.

Tedy funkce považovaná v roce 1950 za jednosměrnou nemusí být za ni považovaná nyní.
Jedna metoda podání formální definice by mohlo být užitím fyzikálního přístupu.

Např. 10^{60}-bitová paměť vždy zůstane nedosažitelnou, protože i kdybychom potřebovali pouze jednu molekulu na bit paměti, její konstrukce by vyžadovala více hmoty než existuje v slunečním systému.

Definice jednosměrné funkce III

Například, výpočet požadující milión instrukcí a 10 000 slov paměti nemohl být v roce 1950 považován za snadný, ale nyní by trval několik sekund na osobním počítači.

Tedy funkce považovaná v roce 1950 za jednosměrnou nemusí být za ni považovaná nyní.
Jedna metoda podání formální definice by mohlo být užitím fyzikálního přístupu.

Např. 10^{60}-bitová paměť vždy zůstane nedosažitelnou, protože i kdybychom potřebovali pouze jednu molekulu na bit paměti, její konstrukce by vyžadovala více hmoty než existuje v slunečním systému.

Podobně, termodynamika nám dává omezení maximálně 10^{70} operací, které lze provést s využitím celkové energie slunce.

Definice jednosměrné funkce III

Například, výpočet požadující milión instrukcí a 10 000 slov paměti nemohl být v roce 1950 považován za snadný, ale nyní by trval několik sekund na osobním počítači.

Tedy funkce považovaná v roce 1950 za jednosměrnou nemusí být za ni považovaná nyní.
Jedna metoda podání formální definice by mohlo být užitím fyzikálního přístupu.

Např. 10^{60}-bitová paměť vždy zůstane nedosažitelnou, protože i kdybychom potřebovali pouze jednu molekulu na bit paměti, její konstrukce by vyžadovala více hmoty než existuje v slunečním systému.

Podobně, termodynamika nám dává omezení maximálně 10^{70} operací, které lze provést s využitím celkové energie slunce.

Nižší úroveň nedosažitelnosti je použití myšlenek výpočetní složitosti.

Definice jednosměrné funkce IV

Nejdříve uvažujme některé z vlastností, které bychom rádi požadovali po jednosměrné funkci.

(I) Výpočet $f(x)$ z x musí být přiměřený: vyjádříme to tím, že f je vypočitatelná v polynomiálně omezené době (říkáme, že $f \in \mathrm{P}$).

Definice jednosměrné funkce IV

Nejdříve uvažujme některé z vlastností, které bychom rádi požadovali po jednosměrné funkci.

(I) Výpočet $f(x)$ z x musí být přiměřený: vyjádříme to tím, že f je vypočitatelná v polynomiálně omezené době (říkáme, že $f \in \mathrm{P}$).

(II) Výpočet f^{-1} nesmí být snadný; budeme tudíž požadovat, že není znám žádný algoritmus pro výpočet f^{-1} v polynomiálně omezené době.

Definice jednosměrné funkce IV

Nejdříve uvažujme některé z vlastností, které bychom rádi požadovali po jednosměrné funkci.

(I) Výpočet $f(x)$ z x musí být přiměřený: vyjádříme to tím, že f je vypočitatelná v polynomiálně omezené době (říkáme, že $f \in \mathrm{P}$).

(II) Výpočet f^{-1} nesmí být snadný; budeme tudíž požadovat, že není znám žádný algoritmus pro výpočet f^{-1} v polynomiálně omezené době.

(III) Třetí podmínka bude tzv. ***upřímnost*** funkce tj., že existuje polynom p splňující $|x| \leq p(|f(x)|)$.

Definice jednosměrné funkce IV

Nejdříve uvažujme některé z vlastností, které bychom rádi požadovali po jednosměrné funkci.

(I) Výpočet $f(x)$ z x musí být přiměřený: vyjádříme to tím, že f je vypočitatelná v polynomiálně omezené době (říkáme, že $f \in \mathrm{P}$).

(II) Výpočet f^{-1} nesmí být snadný; budeme tudíž požadovat, že není znám žádný algoritmus pro výpočet f^{-1} v polynomiálně omezené době.

(III) Třetí podmínka bude tzv. ***upřímnost*** funkce tj., že existuje polynom p splňující $|x| \leq p(|f(x)|)$.

Poslední podmínka je technická podmínka pro vyloučení funkcí jako

$$f(x) = \lceil \log \log x \rceil,$$

která zcela jistě splňuje (I) a (II), ale kterou bychom nemohli obvykle považovat za jednosměrnou funkci.

Definice jednosměrné funkce V

Funkce f splňující (I),(II) a (III) se nazývá slabě jednosměrná funkce.

Definice jednosměrné funkce V

Funkce f splňující (I),(II) a (III) se nazývá slabě jednosměrná funkce.

Příklad. Nechť $\mathbf{I}_k$ značí množinu všech k-bitových přirozených čísel tj.

$$\mathbf{I}_k = \{2^{k-1}, \ldots, 2^k - 1\} \qquad (k = 1, 2, \ldots).$$

Nechť $\mathbf{S}_k = \mathbf{I}_k \times \mathbf{I}_k$ a nechť $f : \mathbf{S}_k \to \mathbf{Z}^+$ je definována jako

$$f(m, n) = m \cdot n.$$

Položíme-li $\mathbf{S} = \bigcup\{\mathbf{S}_k : 1 \leq k < \infty\}$ a rozšíříme-li f na $\mathbf{S}$, získáme slabě jednosměrnou funkci. Přitom v současné době není známo, že by inverzní funkce ležela v $\mathbf{P}$.

Definice jednosměrné funkce VI

Není lehké najít matematickou explicitní definici jednosměrné funkce. Uveďme následující příklad.

Definice jednosměrné funkce VI

Není lehké najít matematickou explicitní definici jednosměrné funkce. Uveďme následující příklad.

Příklad. Buď F šifrovací algoritmus, který zobrazuje zprávu M pomocí klíče K na kryptogram C, $F_K(M) = e(M, K) = C$ a předpokládejme, že $\mathbf{M} \subseteq \mathbf{K}$.

Definice jednosměrné funkce VI

Není lehké najít matematickou explicitní definici jednosměrné funkce. Uveďme následující příklad.

Příklad. Buď F šifrovací algoritmus, který zobrazuje zprávu M pomocí klíče K na kryptogram C, $F_K(M) = e(M, K) = C$ a předpokládejme, že $\mathbf{M} \subseteq \mathbf{K}$.

Změňme trochu tuto funkci. Zafixujme za tímto účelem (ne tajnou) ***zprávu*** M_0 (např. $M_0 = 00\ldots0$); tato "zpráva" se objeví v algoritmu místo obvyklé zprávy, nehraje ale roli variabilní zprávy.

Definice jednosměrné funkce VI

Není lehké najít matematickou explicitní definici jednosměrné funkce. Uveďme následující příklad.

Příklad. Buď F šifrovací algoritmus, který zobrazuje zprávu M pomocí klíče K na kryptogram C, $F_K(M) = e(M, K) = C$ a předpokládejme, že $\mathbf{M} \subseteq \mathbf{K}$.

Změňme trochu tuto funkci. Zafixujme za tímto účelem (ne tajnou) ***zprávu*** M_0 (např. $M_0 = 00\ldots0$); tato "zpráva" se objeví v algoritmu místo obvyklé zprávy, nehraje ale roli variabilní zprávy.

Variabilní zpráva se vloží do algoritmu F na místě klíče. Krátce: uvažujeme funkci

$$f = e(M_0, -) : \mathbf{M} \to \mathbf{C}.$$

Tvrdíme pak, že f je jednosměrná funkce.

Definice jednosměrné funkce VII

Představme si, že Mr. X zná jak M_0 tak i C a chtěl by najít M. V řeči šifrovacího algoritmu F to lze vyjádřit následovně: Mr. X zná sobě odpovídající dvojici zpráva-kryptogram (M_0, C) a chtěl by vypočíst klíč.

Definice jednosměrné funkce VII

Představme si, že Mr. X zná jak M_0 tak i C a chtěl by najít M. V řeči šifrovacího algoritmu F to lze vyjádřit následovně: Mr. X zná sobě odpovídající dvojici zpráva-kryptogram (M_0, C) a chtěl by vypočíst klíč.

Je-li algoritmus F kryptologicky bezpečný, je rezistentní proti tomuto known-plaintext útoku a proto je f jednosměrná funkce.

Definice jednosměrné funkce VII

Představme si, že Mr. X zná jak M_0 tak i C a chtěl by najít M. V řeči šifrovacího algoritmu F to lze vyjádřit následovně: Mr. X zná sobě odpovídající dvojici zpráva-kryptogram (M_0, C) a chtěl by vypočíst klíč.

Je-li algoritmus F kryptologicky bezpečný, je rezistentní proti tomuto known-plaintext útoku a proto je f jednosměrná funkce.

Položme si otázku, zda lze ***matematicky dokázat***, že tato funkce f je jednosměrná. Odpověď zní: ***Ne!***

Definice jednosměrné funkce VII

Představme si, že Mr. X zná jak M_0 tak i C a chtěl by najít M. V řeči šifrovacího algoritmu F to lze vyjádřit následovně: Mr. X zná sobě odpovídající dvojici zpráva-kryptogram (M_0, C) a chtěl by vypočíst klíč.

Je-li algoritmus F kryptologicky bezpečný, je rezistentní proti tomuto known-plaintext útoku a proto je f jednosměrná funkce.

Položme si otázku, zda lze ***matematicky dokázat***, že tato funkce f je jednosměrná. Odpověď zní: ***Ne!***

Matematici nemohli ještě o žádné funkci dokázat, že je jednosměrná.

Definice jednosměrné funkce VIII

To znamená, že neznáme žádnou funkci, jejíž funkční hodnoty lze spočítat v polynomiálně omezené době, ale která při výpočtu funkce inverzní potřebuje exponenciální dobu.

Definice jednosměrné funkce VIII

To znamená, že neznáme žádnou funkci, jejíž funkční hodnoty lze spočítat v polynomiálně omezené době, ale která při výpočtu funkce inverzní potřebuje exponenciální dobu.

Nevíme tedy, zda teoreticky jednosměrné funkce existují. Pro praktické účely mají ale výše popsané funkce dostatečně dobré vlastnosti.

Definice jednosměrné funkce VIII

To znamená, že neznáme žádnou funkci, jejíž funkční hodnoty lze spočítat v polynomiálně omezené době, ale která při výpočtu funkce inverzní potřebuje exponenciální dobu.

Nevíme tedy, zda teoreticky jednosměrné funkce existují. Pro praktické účely mají ale výše popsané funkce dostatečně dobré vlastnosti.

Totiž, kdybychom to uměli, ***uměli bychom dokázat, že*** $P \neq NP$.

Procedura InvMixColumns

V proceduře InvMixColumns dojde ke změně jednotlivých sloupců pole State. Každý bajt se změní na novou hodnotu, která je funkcí všech čtyř bajtů sloupce.

Procedura InvMixColumns

V proceduře InvMixColumns dojde ke změně jednotlivých sloupců pole State. Každý bajt se změní na novou hodnotu, která je funkcí všech čtyř bajtů sloupce.

Stejně jako v proceduře MixColumns se i zde operace provádí modulo $m(x) = x^8 + x^4 + x^3 + x^1 + 1$.

Procedura InvMixColumns

V proceduře InvMixColumns dojde ke změně jednotlivých sloupců pole State. Každý bajt se změní na novou hodnotu, která je funkcí všech čtyř bajtů sloupce.

Stejně jako v proceduře MixColumns se i zde operace provádí modulo $m(x) = x^8 + x^4 + x^3 + x^1 + 1$.

CB	79	6A	90
77	11	1F	C1
8E	5A	A7	5D
3E	7A	5A	77

$$\Rightarrow \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \cdot \begin{bmatrix} CB \\ 77 \\ 8E \\ 3E \end{bmatrix} = \begin{bmatrix} CB \\ CD \\ 96 \\ 9C \end{bmatrix}$$

Tabulka 1: Procedura InvMixColumns

Procedura InvShiftRows

Při proceduře InvShiftRows dojde k posunu v rámci řádků v poli State.

Procedura InvShiftRows

Při proceduře InvShiftRows dojde k posunu v rámci řádků v poli State.

První řádek zůstává stejný, druhý řádek se posune o jednu pozici doprava, třetí řádek o dvě pozice a čtvrtý řádek o tři pozice.

Procedura InvShiftRows

Při proceduře InvShiftRows dojde k posunu v rámci řádků v poli State.

První řádek zůstává stejný, druhý řádek se posune o jednu pozici doprava, třetí řádek o dvě pozice a čtvrtý řádek o tři pozice.

CB	1B	3D	A5
CD	0E	DF	82
96	A8	FB	9E
9C	F5	91	C2

$\Rightarrow$ $\Rightarrow$ $\Rightarrow$ $\Rightarrow$

CB	1B	3D	A5
82	CD	0E	DF
FB	9E	96	A8
F5	91	C2	9C

Tabulka 2: Procedura InvShiftRows

InvSubBytes I

Pro proceduru InvSubBytes byla vytvořena inverzní substituční tabulka InvS-Box.

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

InvSubBytes II

Všechny prvky pole State nahradíme pomocí InvS-Boxu. První čtyři bity prvku označují řádek v tabulce InvS-Box, další čtyři sloupec.

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

InvSubBytes III

Například prvek "D4"ukazuje v S-Boxu na čtrnáctý řádek, čtvrtý sloupek a to je hodnota "19".

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
x	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

InvSubBytes IV

CB	1B	3D	A5
82	CD	0E	DF
FB	9E	96	A8
F5	91	C2	9C

$\Rightarrow$

59	44	8B	29
11	80	D7	EF
63	DF	35	6F
77	AC	A8	1C

Tabulka 3: Procedura InvSubBytes

	y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73