SiCrypt – SAM

wersja 0.1

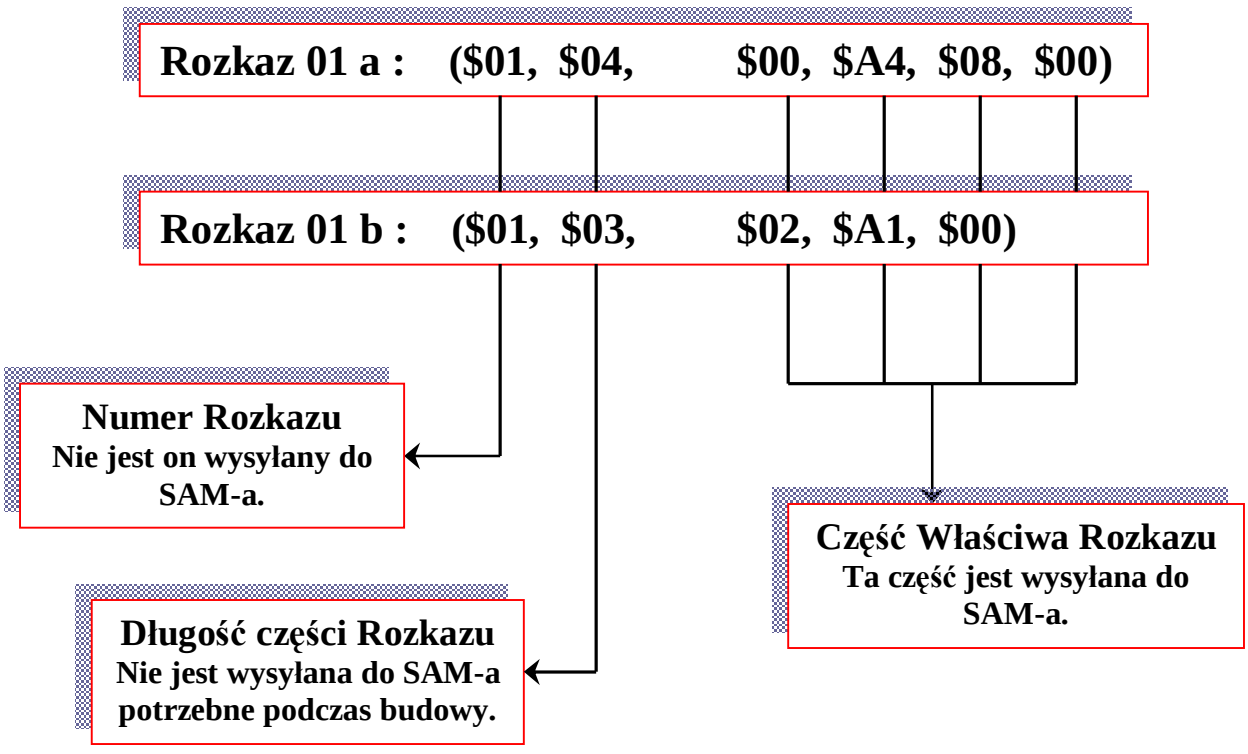
Ogólnie na Początku :

Poniższy opis dotyczy tego co udało mi się do tej pory ustalić na temat komunikacji Płyty Głównej z SAM em. Wszystkie informacje dotyczą Softu Jajka w wersji **3.03** (oczywiście całkiem prawdopodobne jest, że będzie się to zgadzać też w innych softach i warunkach). Jest to całkiem świeży temat (przynajmniej jak dla mnie), więc jest na pewno jeszcze dużo błędów, których nie zauważyłem podczas pisania tego textu i deasemblowania softu.
Text ten należałoby rozpocząć do podania parametrów towarzyszącym transmisji z SAM-em, odbywa się ona z **prędkością 9600 b/s, 2 bity stopu, ramka 8 bitów, parzystość 'Even Parity'**.

Budowa i budowanie Rozkazów :

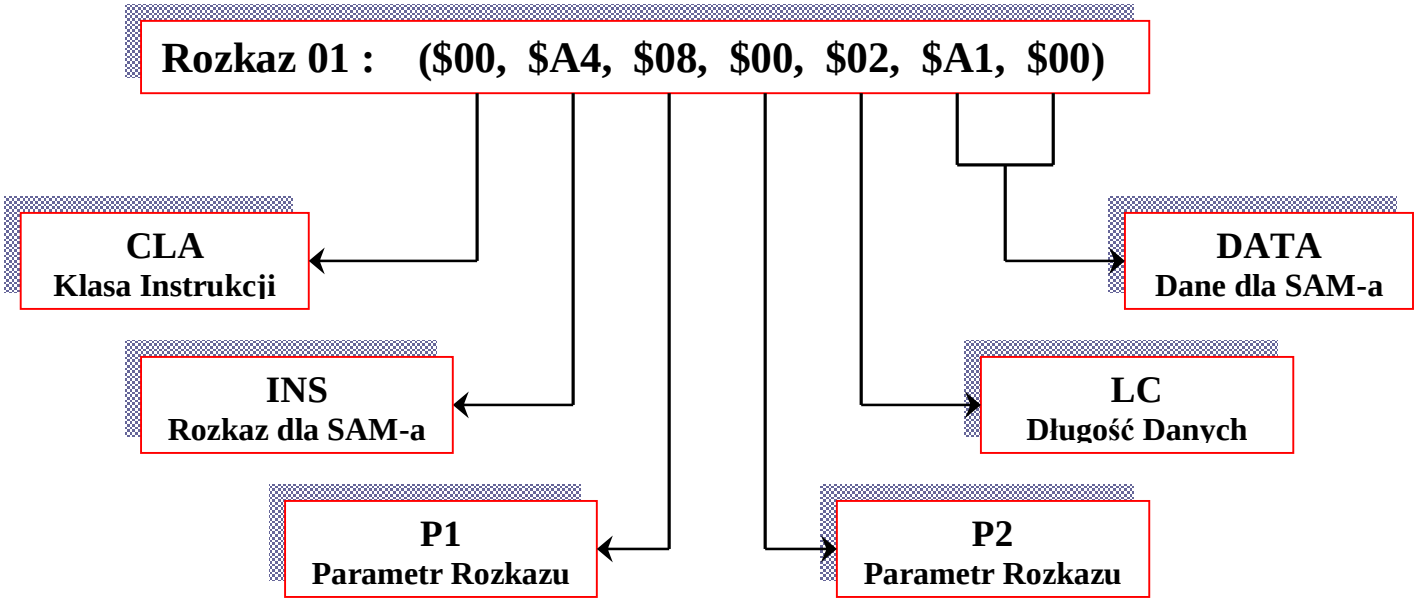
Software z Jajka jest tak zbudowany, że rozkazy (czyli ciąg bajtów wysyłany do SiCrypta bądź w drugą stronę – przyjąłem takie nazewnictwo żeby łatwiej było zrozumieć to co opisuje) są budowane jak gdyby po kawałku, czyli najpierw brana jest pierwsza część rozkazu (z tablicy rozkazów) w tej części zawiera się klasa instrukcji, rozkaz i parametry rozkazu. Następnie dodawana jest druga część rozkazu, zawiera ona w sobie długość danych i dane rozkazu. Ważnym krokiem w tworzeniu rozkazu to sprawdzenie rodzaju danego SAM-a i na podstawie czy jest to SAM 'Czerwony'/'Zielony' lub 'Biały', wyliczana jest suma kontrolna, która dodawana jest na końcu rozkazu i 3 dodatkowe bajty, dodawane są na początku rozkazu (min. przekazana jest cała długość rozkazu) –taka budowa dotyczy Czerwonego/Zielonego a jeśli chodzi o Białego to jest to bardziej skomplikowane i nie są dodawane żadne dodatkowe informacje. Sprawa polega na tym, że w Czerwonym/Zielonym zastosowano protokół **T=1** (asynchroniczny, pół-dupleksowy, blokowy) co mówi nam, że transmisja odbywa się w postaci blokowej i są używane bajty: adres, długość bloku, suma kontrolna itd. Natomiast w Białym użyto protokołu **T=0** (asynchroniczny, pół-dupleksowy, znakowy), oznacza to że transmisja odbywać się będzie w pojedynczych bajtach, nie są używane żadne dodatkowe bajty (długość, suma itp.). Głównie będę starał się opisywać 'Czerwonego'/'Zielonego' SAM-a, ponieważ bardzo mało wiem na temat 'Białego' co oczywiście nie oznacza, że pozostanie tak na wieki. ;)

Poniżej pokaże jak wygląda taki rozkaz w etapie budowy (Czerwony/Zielony) :



Jak widać pierwsze dwa bajty służą tylko i wyłącznie do budowy takich rozkazów (Jajko wie wtedy jaki rozkaz z jakich części skleić i o jakiej długości będzie część właściwa danej połówki). Wygląd takiego sklejonego i prawie gotowego rozkazu pokaże niżej.

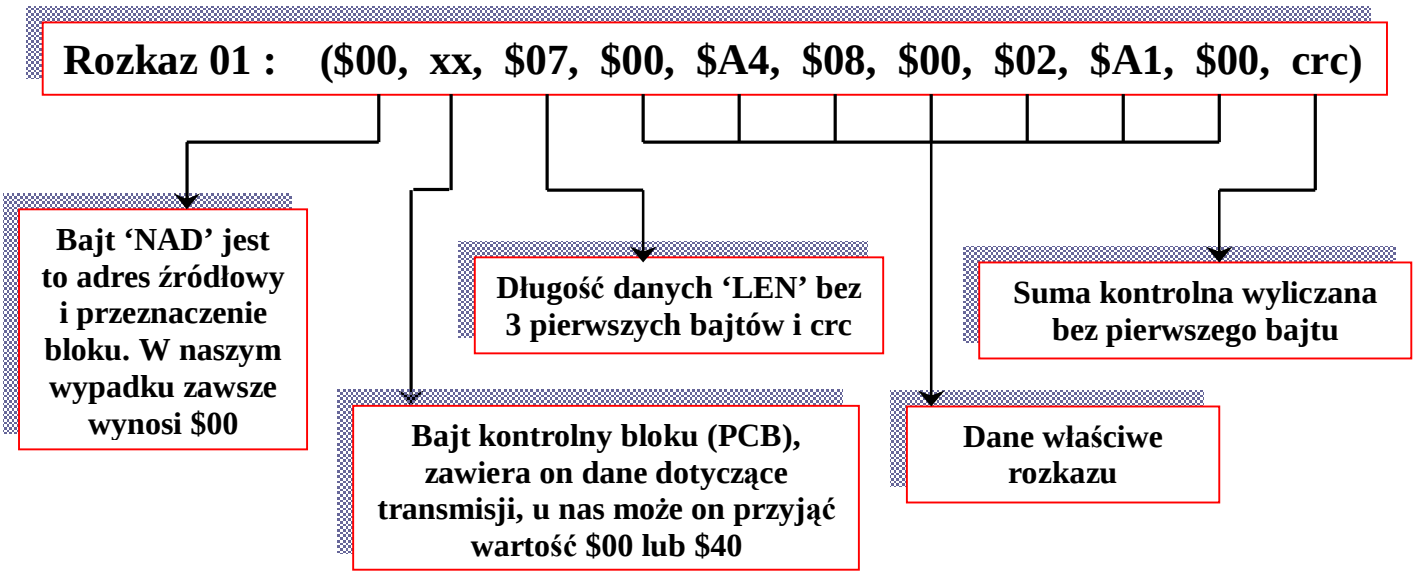
Znaczenie bajtów w rozkazie :



W tym momencie jeśli ktoś potrzebuje bardziej szczegółowych informacji odsyłam do książki „Karta elektroniczna bezpieczny nośnik informacji”.

Właściwy protokół :

Ostatnim krokiem budowania rozkazu przed wysłaniem jest sprawdzenie czy dany rozkaz ma trafić do sama białego czy czerwonego. Informacje o wybraniu odpowiedniego protokołu biorą się z danych uzyskanych z ATR-a, ale o tym później. Jeśli jest to SiCrypt Czerwony to używany jest protokół **T=1** i dane trafiające do sama i z sama będą wyglądały właśnie tak :



Wyliczanie sumy kontrolnej jest bardzo proste w tym przypadku, jest to wynik **xor**-owania kolejnych bajtów w rozkazie zawsze zaczynając od drugiego **(PCB)** a kończąc na ostatnim bajcie danych właściwych rozkazu.

SAM Biały nie potrzebuje takiego dodawania bajtów i sumy kontrolnej, ponieważ używa on protokołu **T=0**. Dane trafiające do Białego są „gołe”, czyli: klasa, instrukcja, parametry itd. Również z takiej komunikacji wynika, że w jednym rozkazie nie mogą przychodzić dane właściwe, które chcemy wysłać bądź uzyskać od sama.

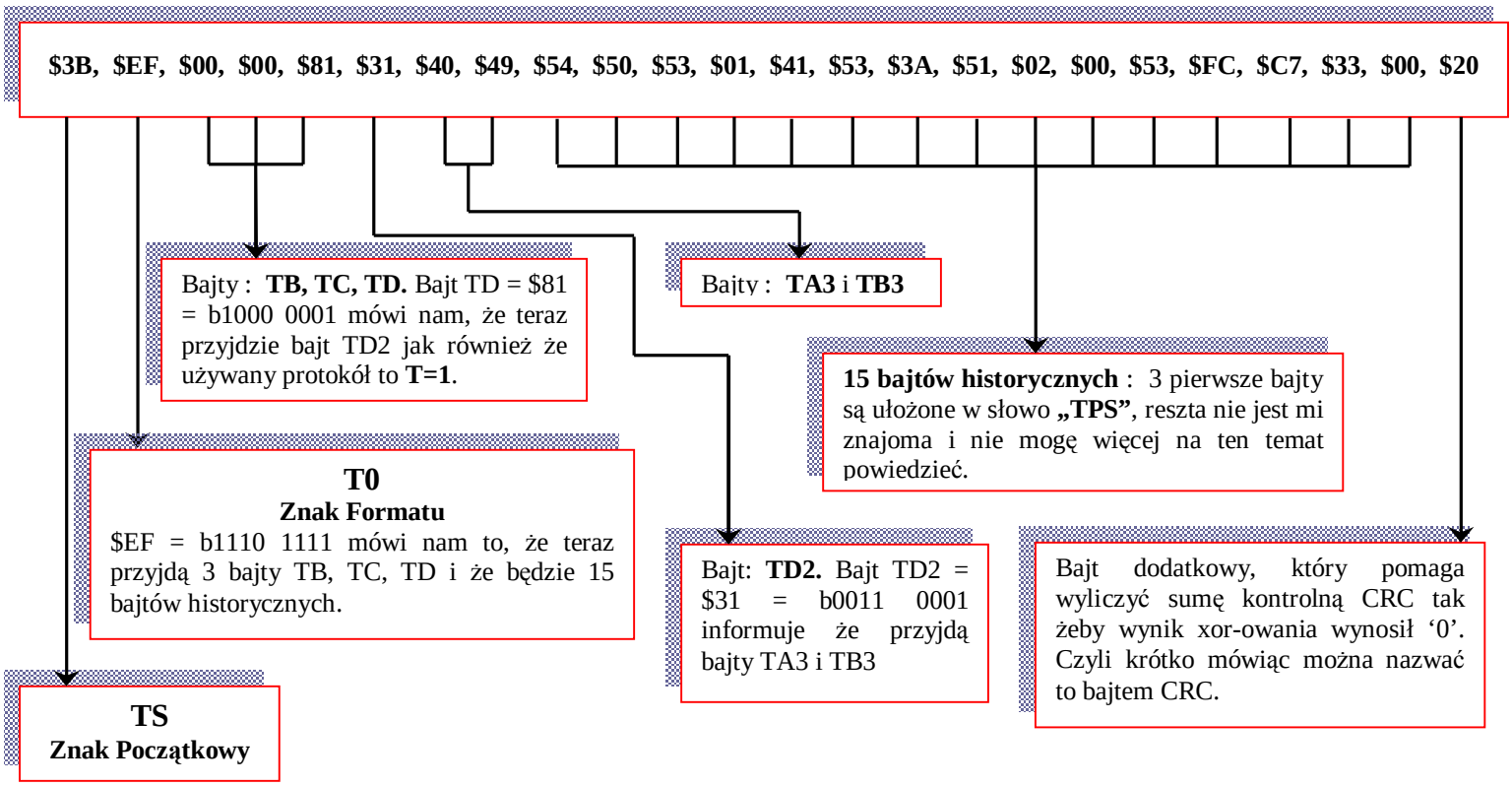
Dane takie będą przesyłane w kolejnej odpowiedzi z sama bądź z płyty głównej, czyli po potwierdzeniu przez jedną ze stron o poprawności otrzymanego rozkazu.

W oprogramowaniu „Jajka” jest przewidziana obsługa jeszcze jednego typu SAM-a, transmisja z takim SiCryptem wygląda trochę inaczej niż normalnie. „Podsłuchując” taką transmisję zauważylibyśmy, że pierwszy bajt ATR-a wynosiłby, **$03** co oczywiście jest zafałszowaniem odczytu, ponieważ takiego typu SAM-a należy potraktować troszkę inaczej. Dlatego po odpowiedniej obróbce tych informacji zaczną nam wychodzić sensowne informacje, czyli pierwszy bajt da nam **$3F**. Więc jak to zostało rozwiązane w rzeczywistości?? Oprogramowanie jest tak napisane, że jeśli pierwszy bajt przychodzący z SAM-a wynosi **$03** to reszta transmisji (w obydwu kierunkach) bajt po bajcie będzie jak gdyby „dekodowany/kodowany”. Tak naprawdę to w bajcie zostaje zamieniony najstarszy bit z najmłodszym itd., czyli najbardziej znaczącym bitem teraz będzie bit wcześniej najmniej znaczący. Na końcu takiej zamiany bitów miejscami wynik jest **NOT**-owany. W obecnej chwili takich typów samów nie używa się jeszcze w automatach, ale kto wie może za jakiś czas ten kawałek kodu będzie wykorzystany. To wszystko, co tu opisałem w związku z tym protokołem jak i inne ciekawe rzeczy są opisane w książce związanej z tą tematyką.

Informacje z ATR-a :

Przyglądając się bajtom ATR-a można wywnioskować m.in., jakie są parametry komunikacji, ile razy można jeszcze wpisać pin, informacje o producencie, itp. Wszystkie te dane są uzależnione w pewien sposób od producenta danej karty, więc da się zauważyć różnicę gołym okiem między samem białym a np. czerwonym.

W tym miejscu opiszę „tyle o ile” co udało mi się zrozumieć. Cały ten galimatias to ATR Czerwonego. Oszczędzę sobie opisu zielonego SAM-a bo wygląda on tak samo.



Opcje Interfejsu :

– TB – (tu wartość **$00**) bajt ten zawiera wartości napięcia i prądu używanego podczas programowania EPROM (dla kontrastu przypomnę, że już od dawna w kartach tego typu nie używa się EPROM-ów tylko EEPROM-ów, czyli tu mamy do czynienia z zabytkiem ;))). Nie jestem w stanie więcej powiedzieć, jakie są to wartości, ponieważ nie dysponuję odpowiednimi normami ISO (dotyczy to raczej większości tych bajtów).

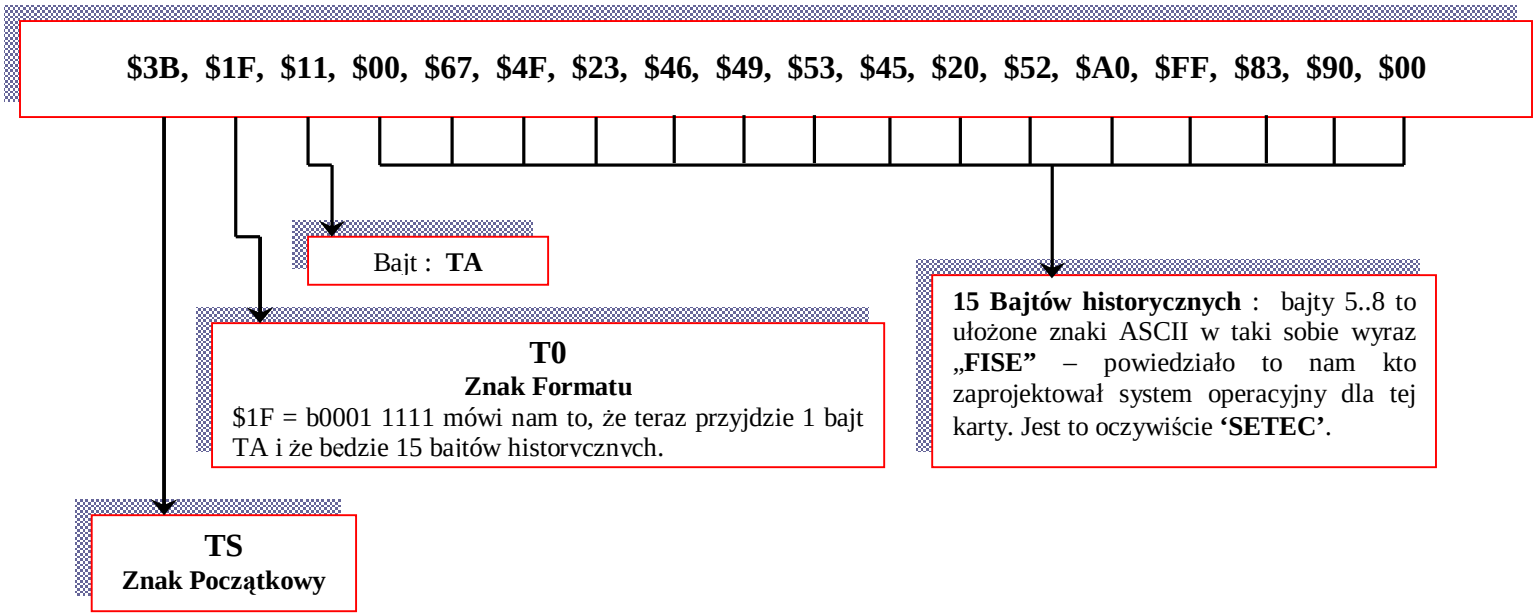
– TC – (tu wartość **$00**) bajt ten zawiera wartość dodatkowego czasu ochronnego między kolejnymi odbieranymi znakami przez SAM-a.

– TA3 – bajt ten zawiera informację o maksymalnej liczbie bajtów wysyłanych w jednym rozkazie do SAM-a (= $40 = #64).

– TB3 – Młodsza część tego bajtu ($9 = 1001) określa czas oczekiwania na jeden znak, natomiast starsza część tego bajtu ($4 = 0100) określa czas oczekiwania na cały blok danych. Dokładnie nie wiem jak z tego wyliczyć czasy.

Mniej/więcej tak to wygląda. Brakuje mi tu bardzo opisu bajtów historycznych, które mogłyby powiedzieć coś więcej na temat: wersji softu itp.

Kontynuując temat ATR-a, postaram się opisać coś na temat „Białego".



Opcje Interfejsu :

Brak bajtu **TD** jest też dla nas informacją, a mianowicie używany protokół przez tą kartę to **T=0**.

– TA – Na podstawie tego bajtu określimy szybkość zegara **F** i współczynnik regulacji szybkości transmisji **D**. W tym wypadku **F = 1** i **D = 1** co daje **FI = 372** i **DI = 1**.

Bajty historyczne.

– T1 – (wartość $00).

– T2 – (wartość $67) Informuje nas, że dane po 10 bajcie są informacją dotyczącą producenta i ich długość.

– T3 – (wartość $4F) Typ użytego układu.

– T4 – (wartość $23) Wersja systemu operacyjnego, czyli ver. 2.3.

– T9 – (wartość $20). Dodatkowy cyfra wersji systemu operacyjnego (i jeszcze coś z chip-em), teraz pełna wersja softu to 2.3.2.
Dodam, że numer wersji mówi mi tylko tyle, że SiCrypt został wyprodukowany, a przynajmniej jego software w latach 96/97 więc w jakich latach został wyprodukowany SAM Czerwony/Zielony ??? A co za tym idzie chyba każdy się może domyślać ;))

– T10 – (wartość $52) Informacja na temat danych twórcy karty i długość danych. (chyba)

– T11 – (wartość $A0) Wielkość buforu.

– T12 – (wartość $FF) To myślę dość istotny bajt, a mówi o tym czy dany SiCrypt miał jakieś błędy podczas pracy. Wartość ta nie powinna wynosić ‘0’, co równe jest jakiejkolwiek odpowiedzi ze strony SAM-a i zakończenia jego żywota. Wartość kryjąca się pod tym bajtem, za każdym razem, gdy wystąpi błąd podczas pracy jest zmniejszana.

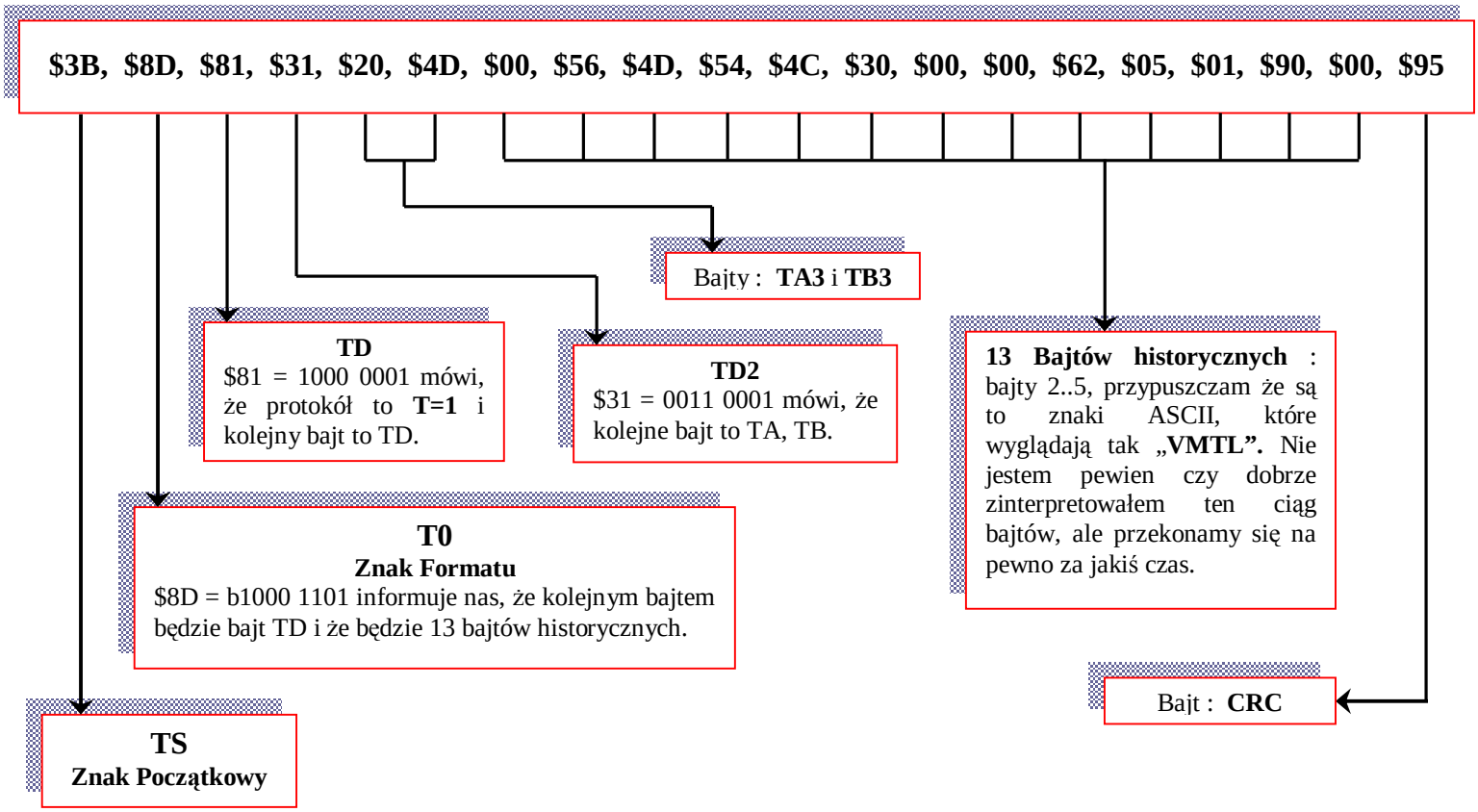
– T13 – (wartość $83) Bajt opisujący istnienie karty, jeśli kolejne bity ustawione są na 1 to :

- bit 0 – Karta została wyprodukowana ale nie sformatowana. (bit = 1 tak)
- bit 1 – Karta została wyprodukowana i sformatowana. (bit = 1 tak)
- bit 2 – Karta została spersonalizowana. (bit = 0 nie)

Jeśli bajt ustawiony jest na $FF czyli wszystkie bity są ustawione na 1, to karta jest martwa :).
Za bardzo nie wiem dokładnie o co chodzi ale tylko tyle zrozumiałem z tego całego syfu.

– T14, T15 – (wartości $90, $00) Słowo statusu karty inaczej SW1, SW2. Wartości te wskazują, że z kartą jest wszystko w porządku, pamięć jest OK. Natomiast, jeśli wartość statusu wynosiłaby: $65, $81 to pamięć karty jest KO.

W tym miejscu postaram się poruszyć temat ATR-a z SiCrypt-a Białego, ale użytego w automacie ASCOM-a.



Opcje Interfejsu :

– TA3 – Jest to maksymalna liczba bajtów wysyłanych w jednym rozkazie do SAM-a ($20 = #32 - domyślnie).

– TB3 – Młodsza część tego bajtu ($D = 1101) określa czas oczekiwania na jeden znak, natomiast starsza część tego bajtu ($4 = 0100) określa czas oczekiwania na cały blok danych. Dokładnie nie wiem jak z tego wyliczyć czasy.

To byłoby na tyle jeśli chodzi delikatny opis ATR-ów. Jeszcze dużo rzeczy pozostaje w tajemnicy, ponieważ nie mamy dostępu do dokumentacji, opisów itd.

Używanie rozkazów :

W jajku zaimplementowane są rozkazy dla każdego SAM-a oddzielnie. Wszystkie są ładnie ponumerowane i aby wybrać odpowiedni zestaw rozkazów ładowane są numerki tych rozkazów do tablicy. Rozkazy pogrupowane są tak jakby na: inicjalizujące, autoryzujące, no i w jednym przypadku na odpowiadające za ładowanie klucza itd.
Struktura tych rozkazów wygląda tak:

Typ rozkazu:	**SAM Czerwony / Zielony**
Inicjalizujące	$01, $02, $03, $04, $00
Autoryzujące	$05, $00
Typ rozkazu:	**SAM jeszcze nie używany w automatach**
Inicjalizujące	$09, $0A, $0C, $0B, $00
Autoryzujące	$0D, $00
Typ rozkazu:	**SAM Biały**
Inicjalizujące	$0F, $10, $11, $12, $13, $14, $15, $00
Autoryzujące	$16, $17, $18, $19, $1A, $1B, $1C, $00
??	$0F, $1D, $1E, $1F, $20, $21, $24, $25, $26, $27, $22, $23, $00
??	$2E, $2F, $2C, $2D, $00
Ładowanie klucza	$24, $25, $28, $27, $29, $2B, $2C, $2D, $24, $25, $26, $27, $22, $23, $00

Rozkazy inicjalizujące m.in. wybierają: odpowiednie katalogi, pliki, klucze. Przesyłają dane z karty Chip-owej do SAM-a, również „pobierają” pytanie z SiCrypta.
Rozkazy autoryzujące wysyłają: dane z karty (najczęściej jest to część z licznikiem) i uzyskaną odpowiedź do SAM-a.
Są jeszcze kombinacje rozkazów, które jak na razie nie wiem dokładnie co robią (dotyczy oczywiście Białego) zaznaczone są „??”.
Jest jeszcze „ładowanie klucza”, o którym zapewne niewiele napiszę.
Drugi SAM jeszcze nie używany to ten, który komunikuje się jak gdyby odwrotnie (czyli zamiana bitów miejscami) i ma w nagłówku ATR-a **$3F.**
Tak poukładane w tej tablicy numery rozkazów oznaczają, że zostaną wykonane właśnie w takiej kolejności i po każdym wykonanym rozkazie konieczna jest odpowiedź ze strony SAM-a.

No teraz pozostało wyjaśnić po kolei, co każdy rozkaz oznacza i w ogóle. Więc zanim to zrobię chcę dodać jeszcze, że rozkaz o numerze $00 to nie rozkaz tylko zakończenie w pewien sposób transmisji. Jeszcze jedna uwaga to, że są rozkazy, które nie zostały wykorzystane w tej tablicy, ale zostały zaimplementowane, je również postaram się omówić.

::

::Czerwony / Zielony::

::

Rozkaz $01 – $00, $A4, $08, $00, $02, $A1, $00
Rozkaz $02 – $00, $50, $01, $00, $05, $A1, $00, $00, $20, $01
Rozkaz $03 – $00, $52, $00, $00, $0B, $00, 'DK8', $71, $01
Rozkaz $04 – $00, $54, $00, $00, $06
Rozkaz $05 – $00, $90, $00, $10, $0A, 'IK8', 'OK2'
Rozkaz $06 – $00, $90, $80, $10, $0A, 'IK8', 'OK2'
Rozkaz $07 – $00, $90, $01, $10, $0A, 'IK8', 'OK2'

::

:::Nieużywany:::

::

Rozkaz $09 – $A0, $A4, $00, $00, $02, $7F, $00
Rozkaz $0A – $A0, $A4, $00, $00, $02, $00, $01
Rozkaz $0B – $A0, $84, $00, $00, $10
Rozkaz $0C – $80, $52, $00, $00, $08, 'DK8'
Rozkaz $0D – $80, $82, $00, $00, $12, 'DK16', 'OK2'

::

:::Biały::

::

Rozkaz $0F – $81, $A4, $08, $00, $04, $DF, $AB, $EF, $FF
Rozkaz $10 – $81, $AE, $01, $00, $02, $82, 'TE1'
Rozkaz $11 – 'OD5'
Rozkaz $12 – '4OB1', 'DKx', $90, $00
Rozkaz $13 – 'OD5'
Rozkaz $14 – '2OB1'
Rozkaz $15 – 'OD6'
Rozkaz $16 – $61, $02
Rozkaz $17 – 'OD5'
Rozkaz $18 – '2OB1', 'OK2', $90, $00
Rozkaz $19 – 'OD5'
Rozkaz $1A – '2OB1'
Rozkaz $1B – 'OD28'
Rozkaz $1C – $90, $00
Rozkaz $1D – $81, $AE, $21, $00, $00
Rozkaz $1E – 'OD5'
Rozkaz $1F – $5A
Rozkaz $20 – 'OD39'
Rozkaz $21 – $90, $00
Rozkaz $22 – $81, $B0, $00, $00, $0A
Rozkaz $23 – 'OD13'
Rozkaz $24 – $81, $A4, $02, $00, $02
Rozkaz $25 – 'OD1'
Rozkaz $26 – $00, $21
Rozkaz $27 – (przerwa w nadawaniu ??).
Rozkaz $28 – $EF, $FF
Rozkaz $29 – $81, $AE, $FD, $00, $22
Rozkaz $2B – $70, 'KS33'
Rozkaz $2C – 'OD5'
Rozkaz $2D – $90, $00
Rozkaz $2E – $81, $A4, $08, $0C, $04, $DF, $AB, $EF, $FF
Rozkaz $2F – $81, $AE, $26, $00, $07, 'CC7'

::

::

Jest jeszcze rozkaz $08, który nie jest wywołany po numerku tylko w momencie, gdy długość rozkazu wysyłanego do SAM-a jest mniejsza od 5 (dotyczy tylko samów Czerwonego/Zielonego i nie używanego) a wygląda tak.:

Rozkaz $08 – $A0, $C0, $00, $00, $00

Legenda :

‘DK8’ - (Dane z Karty) 8 pierwszych bajtów z karty, czyli nagłówek, emisja, seria, numer, itd.
‘DK16’ - (Dane z Karty) 16 pierwszych bajtów z karty, czyli wszystko to, co wyżej tyle, że jeszcze dochodzą impulsy inaczej licznik 8 bajtów (dodam tylko, że są one odwrócone, czyli w bajcie zamienione jest bit młodszy ze starszym itd.).
‘DKx’ - (Dane z Karty) Tutaj długość danych z karty jest określona w danych poprzedzających ten rozkaz otrzymany z SiCrypt-a na pozycji 7.
‘IK8’ - (Impulsy z Karty) 8 bajtów z karty, część z licznikiem, czyli impulsami.
‘OK2’ - (Odpowiedź z Karty) 2 bajty odpowiedzi z karty.
‘TE1’ - (Tablica Emisji) 1 bajt „wyciągnięty” z tablicy emisji na podstawie emisji z karty.
‘4OB1’ - (Odebrany Bajt) 1 bajt „wyciągnięty” z danych otrzymanych z SAM-a poprzedzających ten rozkaz na pozycji 4.
‘2OB1’ - (Odebrany Bajt) wszystko to samo, co wyżej tyle, że na pozycji 2.
‘KS33’ - (Klucz dla SAM-a) 33 bajty klucza kryptograficznego.
‘CC7’ - (Config Chip) 7 bajtów niewiadomo za bardzo do czego są. Na razie są ładowane domyślnie z softu ale w przyszłości mogą ulec zmianie, ponieważ jeśli zostanie załadowany do automatu plik (**File Config Chip**) to wprowadzona wartość będzie pochodziła z tego pliku. Na dzień dzisiejszy te bajty wyglądają tak: (**$30, $30, $30, $30, $30, $31, $00**) = ASCII (**000001**)
‘ODx’ - (Odbierz Dane) x – określa ile bajtów danych, automat oczekuje od SAM-a.

Jak łatwo zauważyć SAM Czerwony/Zielony i ten który jeszcze nie jest używany w automatach, nie posiadają możliwości ładowania zdalnie kluczy, jest to uwarunkowane softwarem, który został przygotowany z tą opcją tylko dla SAM-a Białego. Budowanie rozkazów dla trzech pierwszych SAM-ów na tym się nie kończy, zresztą opisałem to już wcześniej, chodzi o dodanie trzech pierwszych bajtów i CRC.
Jeśli chodzi o rozkazy $06 i $07 nie są używane w komunikacji z SAM-em Czerwonym/Zielonym ale są zaimplementowane i są to rozkazy autoryzujące.

Na zakończenie :

Brakuje mi dużo informacji na temat rozkazów, czyli dokładnie, co dany rozkaz robi. Mniej/więcej można się domyślać, że zawsze na początku (np. rozkaz $01) wybierany jest katalog, następnie wybierany jest plik z odpowiednimi kluczami a na samym końcu wysyłane są dane do odpowiedniego algorytmu. Nie mam dokumentacji od danego typu SAM-a, więc w większości można się tylko domyślać. W wielu przypadku normy ISO i różne inne ogólne opisy są nie przydatne, ponieważ producenci tych SAM-ów odbiegali od normowych rygorów ;)) a co za tym idzie, jedynym źródłem wiarygodnej informacji jest dokumentacja od konkretnego typu i wersji SiCrypt-a.

Na tym etapie zakończę ten nie wielki opis SAM-ów i na zakończenie zamieściłem kilka poglądowych logów z komunikacji Silverka z SiCrypt-em.

--- Czerwony ---
=== 18 6A 75 === 25 31 09 ===

Karta nr.: 6321059675 imp. 24 – Włożenie karty do czytnika Silverka	
SiCrypt - ATR -	$3B, $EF, $00, $00, $81, $31, $40, $49, $54, $50, $53, $01, $41, $53, $3A, $51, $02, $00, $53, $FC, $C7, $33, $00, $20
Czytnik - Rozk. 01 -	$00, $00, $07, $00, $A4, $08, $00, $02, $A1, $00, $08
SiCrypt - OK -	$00, $00, $02, $90, $00, $92
Czytnik - Rozk. 02 -	$00, $40, $0A, $00, $50, $01, $00, $05, $A1, $00, $00, $22, $01, $9C
SiCrypt - OK -	$00, $40, $02, $90, $00, $D2
Czytnik - Rozk. 03 -	$00, $00, $10, $00, $52, $00, $00, $0B, $00, $05, $F4, $26, $FC, $D2, $1C, $E9, $1B, $71, $01, $2E
SiCrypt - OK -	$00, $00, $02, $90, $00, $92
Czytnik - Rozk. 04 -	$00, $40, $05, $00, $54, $00, $00, $08, $19
SiCrypt - Pytanie -	$00, $40, $0A, $9E, $89, $75, $CB, $BE, $94, $A6, $13, $90, $00, $EC
Czytnik - Rozk. 06 -	$00, $00, $0F, $00, $90, $80, $10, $0A, $00, $00, $00, $E0, $00, $7F, $FF, $3F, $76, $E9, $C5
SiCrypt - OK -	$00, $00, $02, $90, $00, $92

Karta nr.: 6321059675 imp. 23 – Skasowanie 1 impulsu w czytniku Silverka	
SiCrypt - ATR -	$3B, $EF, $00, $00, $81, $31, $40, $49, $54, $50, $53, $01, $41, $53, $3A, $51, $02, $00, $53, $FC, $C7, $33, $00, $20
Czytnik - Rozk. 01 -	$00, $00, $07, $00, $A4, $08, $00, $02, $A1, $00, $08
SiCrypt - OK -	$00, $00, $02, $90, $00, $92
Czytnik - Rozk. 02 -	$00, $40, $0A, $00, $50, $01, $00, $05, $A1, $00, $00, $22, $01, $9C
SiCrypt - OK -	$00, $40, $02, $90, $00, $D2
Czytnik - Rozk. 03 -	$00, $00, $10, $00, $52, $00, $00, $0B, $00, $05, $F4, $26, $FC, $D2, $1C, $E9, $1B, $71, $01, $2E
SiCrypt - OK -	$00, $00, $02, $90, $00, $92
Czytnik - Rozk. 04 -	$00, $40, $05, $00, $54, $00, $00, $08, $19
SiCrypt - Pytanie -	$00, $40, $0A, $6B, $75, $44, $2F, $4F, $4D, $8E, $0E, $90, $00, $2D
Czytnik - Rozk. 06 -	$00, $00, $0F, $00, $90, $80, $10, $0A, $00, $00, $00, $C0, $FE, $7F, $FF, $1F, $02, $9B, $3D
SiCrypt - OK -	$00, $00, $02, $90, $00, $92

--- Czerwony ---
=== 18 6A 75 === 1E 09 0E ===

Karta nr.: 6321059675 imp. 23 – Włożenie karty do czytnika Silverka	
SiCrypt - ATR -	$3B, $EF, $00, $00, $81, $31, $40, $49, $54, $50, $53, $01, $41, $53, $3A, $51, $02, $00, $53, $FC, $C7, $33, $00, $20
Czytnik - Rozk. 01 -	$00, $00, $07, $00, $A4, $08, $00, $02, $A1, $00, $08
SiCrypt - OK -	$00, $00, $02, $90, $00, $92
Czytnik - Rozk. 02 -	$00, $40, $0A, $00, $50, $01, $00, $05, $A1, $00, $00, $22, $01, $9C
SiCrypt - OK -	$00, $40, $02, $90, $00, $D2
Czytnik - Rozk. 03 -	$00, $00, $10, $00, $52, $00, $00, $0B, $00, $05, $F4, $26, $FC, $D2, $1C, $E9, $1B, $71, $01, $2E
SiCrypt - OK -	$00, $00, $02, $90, $00, $92
Czytnik - Rozk. 04 -	$00, $40, $05, $00, $54, $00, $00, $08, $19
SiCrypt - Pytanie -	$00, $40, $0A, $F7, $A3, $02, $93, $30, $D7, $47, $5C, $90, $00, $E3
Czytnik - Rozk. 06 -	$00, $00, $0F, $00, $90, $80, $10, $0A, $00, $00, $00, $C0, $FE, $7F, $FF, $1F, $60, $3B, $FF
SiCrypt - OK -	$00, $00, $02, $90, $00, $92

Karta nr.: 6321059675 imp. 22 – Skasowanie 1 impulsu w czytniku Silverka	
SiCrypt - ATR -	$3B, $EF, $00, $00, $81, $31, $40, $49, $54, $50, $53, $01, $41, $53, $3A, $51, $02, $00, $53, $FC, $C7, $33, $00, $20
Czytnik - Rozk. 01 -	$00, $00, $07, $00, $A4, $08, $00, $02, $A1, $00, $08
SiCrypt - OK -	$00, $00, $02, $90, $00, $92
Czytnik - Rozk. 02 -	$00, $40, $0A, $00, $50, $01, $00, $05, $A1, $00, $00, $22, $01, $9C
SiCrypt - OK -	$00, $40, $02, $90, $00, $D2
Czytnik - Rozk. 03 -	$00, $00, $10, $00, $52, $00, $00, $0B, $00, $05, $F4, $26, $FC, $D2, $1C, $E9, $1B, $71, $01, $2E
SiCrypt - OK -	$00, $00, $02, $90, $00, $92
Czytnik - Rozk. 04 -	$00, $40, $05, $00, $54, $00, $00, $08, $19
SiCrypt - Pytanie -	$00, $40, $0A, $1C, $B0, $FC, $CE, $1E, $BF, $16, $90, $90, $00, $63
Czytnik - Rozk. 06 -	$00, $00, $0F, $00, $90, $80, $10, $0A, $00, $00, $00, $C0, $FC, $7F, $FF, $1F, $4C, $0C, $E6
SiCrypt - OK -	$00, $00, $02, $90, $00, $92

--- Zielony ---

=== 18 66 B9 === 1E 21 1F ===

Karta nr.: 6321059675 imp. 23 – Włożenie karty do czytnika Silverka	
SiCrypt - ATR -	**$3B, $EF, $00, $00, $81, $31, $40, $49, $54, $50, $53, $01, $41, $53, $3A, $51, $02, $00, $53, $FC, $C7, $33, $00, $20**
Czytnik - Rozk. 01 -	**$00, $00, $07, $00, $A4, $08, $00, $02, $A1, $00, $08**
SiCrypt - OK -	**$00, $00, $02, $90, $00, $92**
Czytnik - Rozk. 02 -	**$00, $40, $0A, $00, $50, $01, $00, $05, $A1, $00, $00, $22, $01, $9C**
SiCrypt - OK -	**$00, $40, $02, $90, $00, $D2**
Czytnik - Rozk. 03 -	**$00, $00, $10, $00, $52, $00, $00, $0B, $00, $05, $F4, $26, $FC, $D2, $1C, $E9, $1B, $71, $01, $2E**
SiCrypt - OK -	**$00, $00, $02, $90, $00, $92**
Czytnik - Rozk. 04 -	**$00, $40, $05, $00, $54, $00, $00, $08, $19**
SiCrypt - Pytanie -	**$00, $40, $0A, $F7, $A3, $02, $93, $30, $D7, $47, $5C, $90, $00, $E3**
Czytnik - Rozk. 06 -	**$00, $00, $0F, $00, $90, $80, $10, $0A, $00, $00, $00, $C0, $FE, $7F, $FF, $1F, $60, $3B, $FF**
SiCrypt - OK -	**$00, $00, $02, $90, $00, $92**
Karta nr.: 6321059675 imp. 22 – Skasowanie 1 impulsu w czytniku Silverka	
SiCrypt - ATR -	**$3B, $EF, $00, $00, $81, $31, $40, $49, $54, $50, $53, $01, $41, $53, $3A, $51, $02, $00, $53, $FC, $C7, $33, $00, $20**
Czytnik - Rozk. 01 -	**$00, $00, $07, $00, $A4, $08, $00, $02, $A1, $00, $08**
SiCrypt - OK -	**$00, $00, $02, $90, $00, $92**
Czytnik - Rozk. 02 -	**$00, $40, $0A, $00, $50, $01, $00, $05, $A1, $00, $00, $22, $01, $9C**
SiCrypt - OK -	**$00, $40, $02, $90, $00, $D2**
Czytnik - Rozk. 03 -	**$00, $00, $10, $00, $52, $00, $00, $0B, $00, $05, $F4, $26, $FC, $D2, $1C, $E9, $1B, $71, $01, $2E**
SiCrypt - OK -	**$00, $00, $02, $90, $00, $92**
Czytnik - Rozk. 04 -	**$00, $40, $05, $00, $54, $00, $00, $08, $19**
SiCrypt - Pytanie -	**$00, $40, $0A, $1C, $B0, $FC, $CE, $1E, $BF, $16, $90, $90, $00, $63**
Czytnik - Rozk. 06 -	**$00, $00, $0F, $00, $90, $80, $10, $0A, $00, $00, $00, $C0, $FC, $7F, $FF, $1F, $4C, $0C, $E6**
SiCrypt - OK -	**$00, $00, $02, $90, $00, $92**

--- Biały z ASCOM-a ---

Włożenie karty do czytnika Silverka	
SiCrypt - ATR -	**$3B, $8D, $81, $31, $20, $4D, $00, $56, $4D, $54, $4C, $30, $00, $00, $62, $05, $01, $90, $00, $95**
Czytnik - Rozk. 01 -	**$00, $00, $07, $00, $A4, $08, $00, $02, $A1, $00, $08**
SiCrypt	**$00, $81, $00, $81**
Czytnik - Rozk. 02 -	**$00, $00, $0A, $00, $50, $01, $00, $05, $A1, $00, $00, $22, $01, $DC**
SiCrypt	**$00, $81, $00, $81**
Czytnik - Rozk. 03 -	**$00, $00, $10, $00, $52, $00, $00, $0B, $00, $05, $F4, $26, $F4, $09, $97, $09, $0F, $71, $01, $82**

W tym momencie transmisja się kończy, ponieważ SiCrypt nie zrozumiał rozkazu a czytnik czekał na odpowiedź z sama, a co śmieszniejsze to, że ten SAM od początku nie rozumiał danych rozkazów i wysyłał $81, $00 co prawdopodobnie oznacza błędny rozkaz, czytnik oczywiście ignorował rozkazy i wysyłał „swoje”. Próba została powtórzona jeszcze raz przez czytnik czyli: reset i po kolei te same rozkazy po drugiej nie udanej znowu próbie automat stwierdził że karta jest nie czytelna ;))

Ogłoszenia drobne :

APELUJĘ.
- Jeśli ktokolwiek ma jakiekolwiek informacje na temat SAM-ów, to proszę o kontakt.
- Interesują mnie jakiekolwiek dokumentacje na temat SiCryptów.
- Normy **ISO** od numeru **7816-1** do **7816-7**.
- Potrzebuję zlogowanej komunikacji SAM-ów Czerwonego, Zielonego, Białego i Białego z ASCOM-a, w automatach Jajko i ASCOM.
- Jeśli znalazłeś błędy w opisie to proszę napisz do mnie.
- Jeśli posiadasz Białego SAM-a innego niż **1.36** to napisz do mnie.

Xzbir@poczta.onet.pl
http://phreak.hack.pl