Architektura Bezpieczeństwa dla Systemu ROVERS

Artur Skrajnowski

Opiekun: dr inż. Jarosław Domaszewicz

Współpraca: prof. dr hab. Zbigniew Kotulski

Plan prezentacji

- Sieci sensorowe
- System ROVERS
- Bezpieczeństwo informacji w ROVERS
- Aktualnie istniejące rozwiązania
- Zaprojektowane rozwiązanie
- Dalsze możliwości rozwoju

Plan prezentacji

- Sieci sensorowe
  - **Charakterystyka**
  - **Zastosowania**
- System ROVERS
- Bezpieczeństwo informacji w ROVERS
- Aktualnie istniejące rozwiązania
- Zaprojektowane rozwiązanie
- Dalsze możliwości rozwoju

Sieci sensorowe - cechy

- Węzły o małej wydajności zawierające podstawowe moduły: procesor, pamięć, radio, sensory
- Czas życia węzłów ograniczony zapasem energii (najczęściej nieodnawialnej)
- Sieć o dynamicznej topologii (mobilność węzłów, awarie)
- Sieć powinna działać bez ingerencji użytkownika
- Typowo: pole sensorowe + stacja bazowa analizująca pomiary

Sieci sensorowe - zastosowania

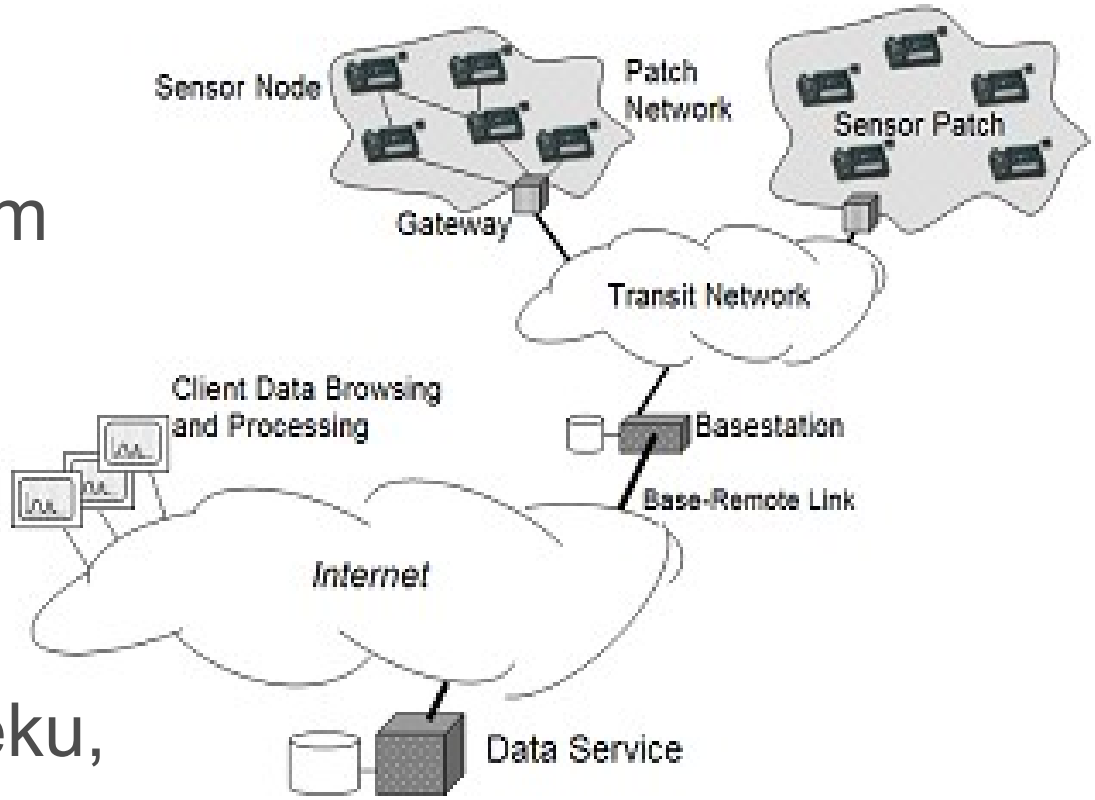
- Monitorowanie, Kontrola, Śledzenie
- Używane tam, gdzie inne metody okazują się niedostępne lub za drogie
- Przykłady zastosowania:
  - **Wykrywanie pożarów**
  - **Śledzenie ruchu wojsk**
  - **Monitorowanie środowiska**
  - **Monitorowanie składu atmosfery**
  - **„Inteligentne przestrzenie”**

Sieci sensorowe – przykład (1/2)

- Monitorowanie ośrodka lęgowego petreli wielkich położonego na Great Duck Island
  - **Wykorzystanie nor na gniazda**
  - **Warunki panujące w gnieździe i poza nim**
  - **Wpływ gęstości gniazd na warunki panujące w gniazdach**
- Okres lęgowy trwa 9 miesięcy
- 15 min. wizyta człowieka w kolonii zwiększa umieralność piskląt o 20%

Sieci sensorowe – przykład (2/2)

- Sensory umieszczone przed okresem lęgowym
- Mierzono temperaturę, wilgotność, naświetlenie, poziom dźwięku, ciśnienie



Plan prezentacji

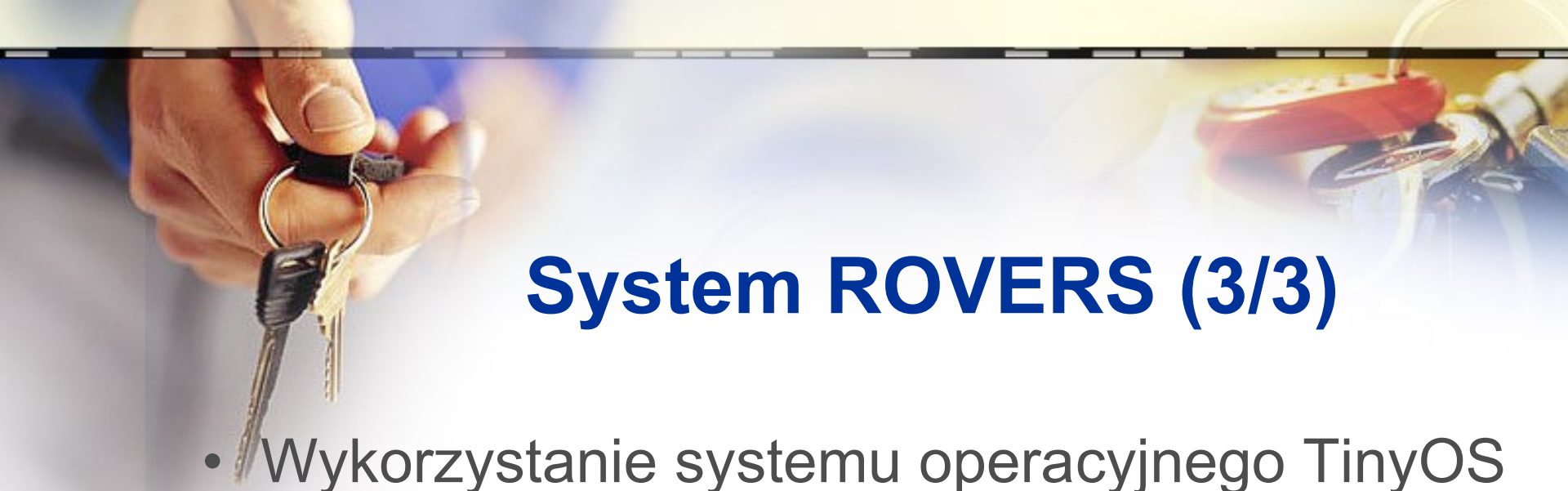
- Sieci sensorowe
- System ROVERS
  - **Przeznaczenie**
  - **Charakterystyka**
  - **Przykład aplikacji**
- Bezpieczeństwo informacji w ROVERS
- Aktualnie istniejące rozwiązania
- Zaprojektowane rozwiązanie
- Dalsze możliwości rozwoju

System ROVERS (1/3)

- Projekt zapoczątkowany przez grupę Mobile and Embedded Application Group (MEAG) pod przewodnictwem dr inż. Jarosława Domaszewicza
- Aktualnie w fazie implementacji na symulator sieci sensorowej
- Zawiera istotne różnice w porównaniu z typową siecią sensorową

System ROVERS (2/3)

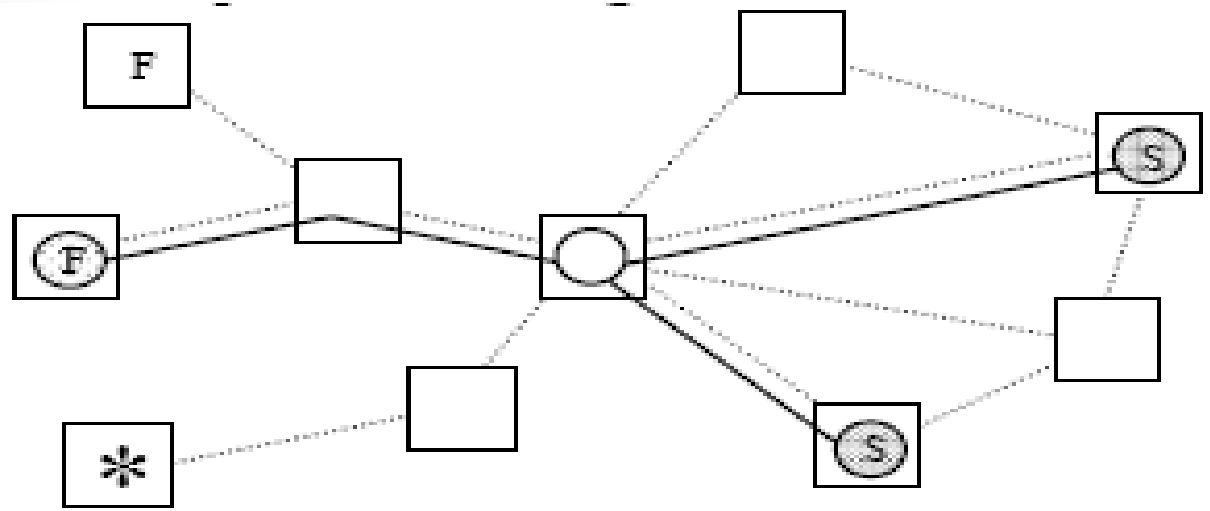
- System przeznaczony na domowe sieci sensorowe Wireless Sensor-Actuator Network
- Węzły sieci (sensory) umieszczone w przedmiotach codziennego użytku:
  - **Sprzęt AGD, RTV**
  - **Oświetlenie, wentylacja**
  - **Inne...**
- Sieć p2p – każdy węzeł jest równorzędny, zarządzanie siecią rozproszone – brak stacji bazowej



System ROVERS (3/3)

- Wykorzystanie systemu operacyjnego TinyOS
- ROVERS jest warstwa middleware ułatwiającą pisanie aplikacji na WSAN
- Mikro-agenci – prymitywy programistyczne działające na fizycznych węzłach, współpracujące ze sobą realizując pewną aplikację
- Mobilność mikro-agentów
- Wprowadzanie nowych aplikacji do systemu

Przykład aplikacji - KitchenAirManager



- □ - fizyczny węzeł
- F – wiatrak, S – czujnik dymu
- ☼ - pierwotny mikro-agent aplikacji

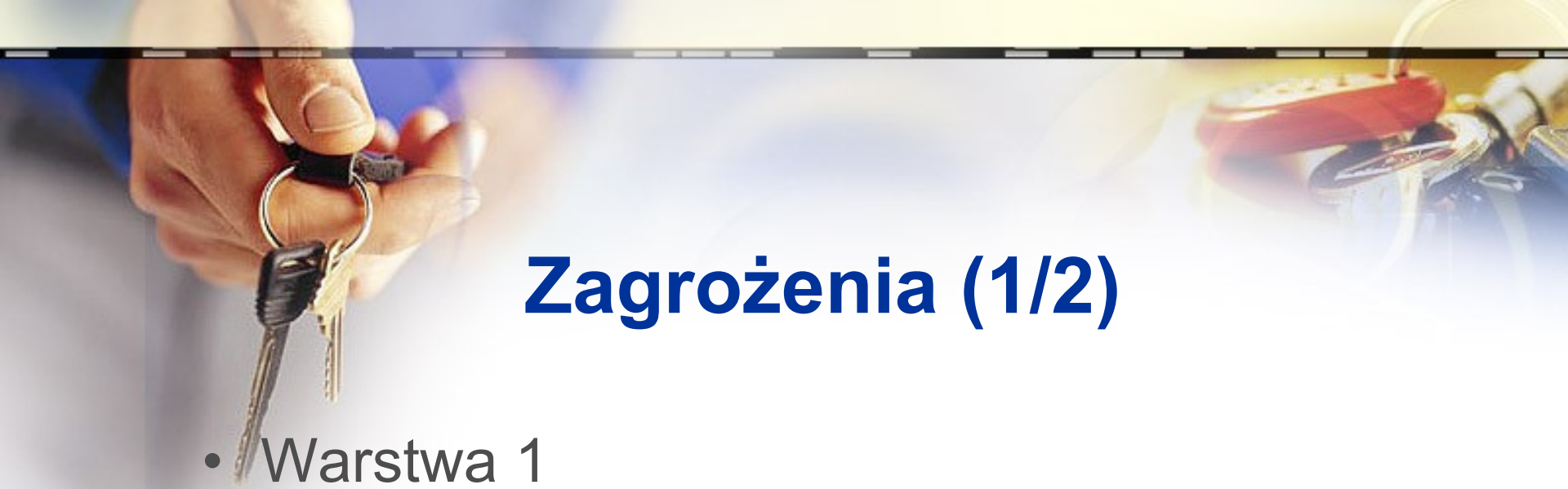
Plan prezentacji

- Sieci sensorowe
- System ROVERS
- Bezpieczeństwo informacji w ROVERS
  - **Czy kryptografia potrzebna ?**
  - **Zagrożenia**
- Aktualnie istniejące rozwiązania
- Zaprojektowane rozwiązanie
- Dalsze możliwości rozwoju

Kryptografia potrzebna ?

- System oparty o komunikację radiową wykorzystującą wspólne medium
- Informacje przesyłane nie są krytyczne, ale ataki mogą być uciążliwe
- TinyOS nie ma w sobie komponentów zapewniających bezpieczeństwo

- Wniosek => Zapewnienie bezpieczeństwa informacji jest konieczne



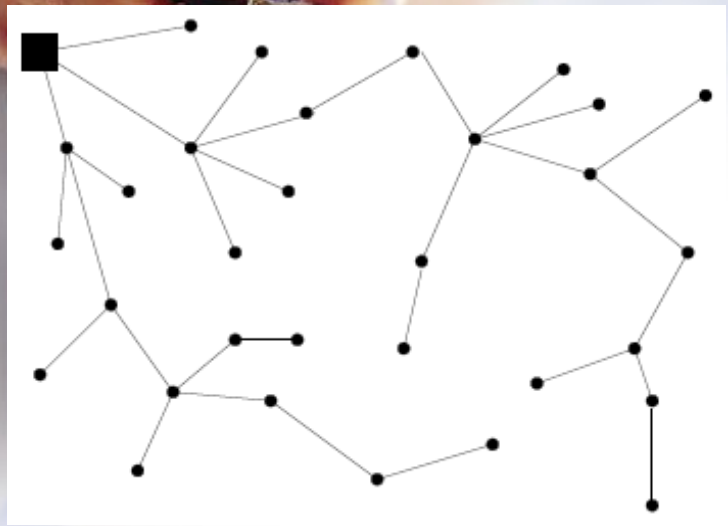
Zagrożenia (1/2)

- Warstwa 1
  - **Zagłuszanie**
  - **„Sleep depravation attack”**
  - **Kradzież, uszkodzenie urządzenia**
- Warstwa 2
  - **Podsłuch**
  - **Zmiana ramek wysyłanych przez węzły**
  - **Wprowadzenie błędnych ramek**
  - **Przekazywanie tylko wybranych ramek**

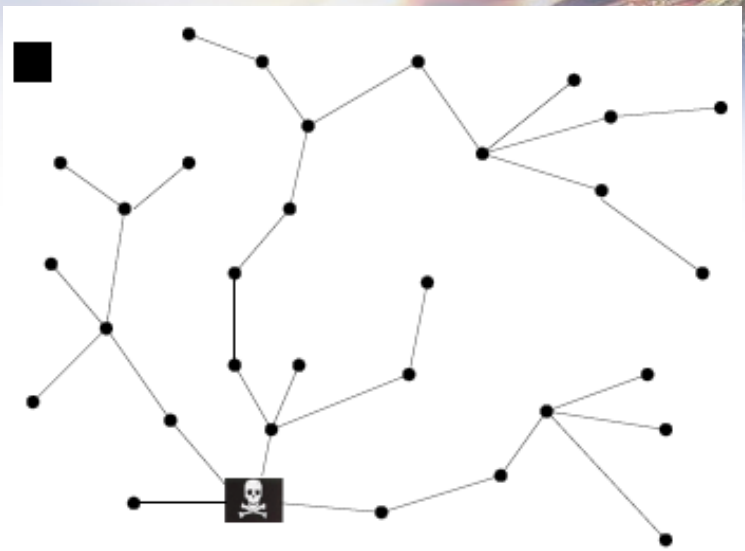
Zagrożenia (2/2)

- Warstwa 3 – wprowadzanie błędnej informacji routingowej
  - **Sinkhole, Wormhole**
  - **Hello flood**
- Warstwy wyższe – wprowadzenie złośliwego kodu do sieci
  - **Mikro-agenci**
  - **Aplikacje**

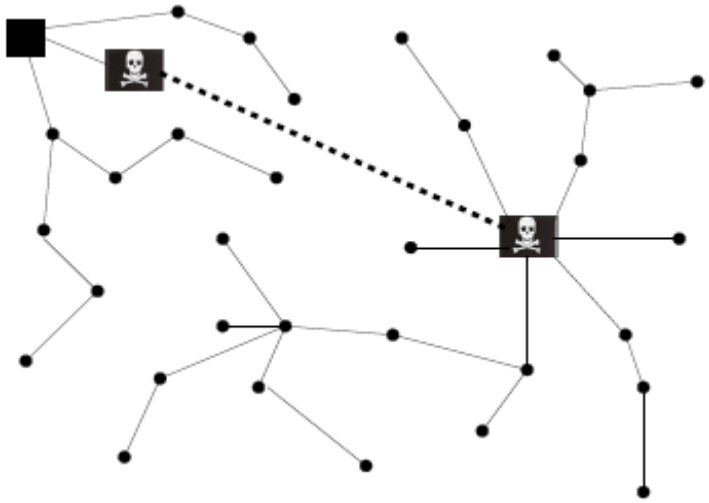
Topologia sieci



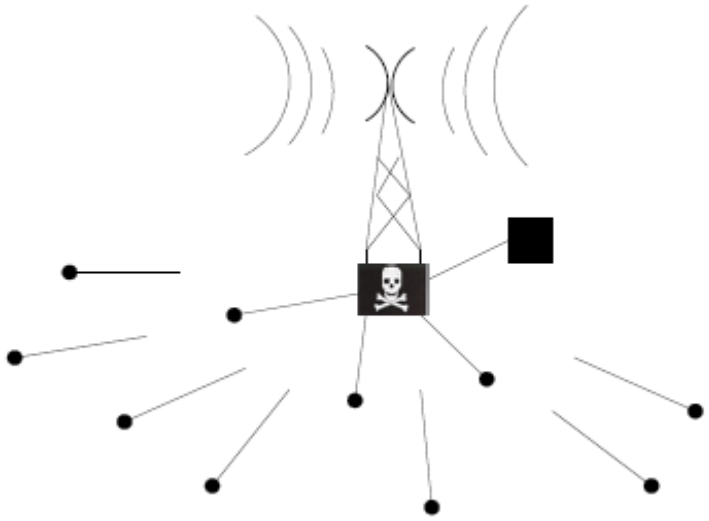
Sinkhole



Wormhole

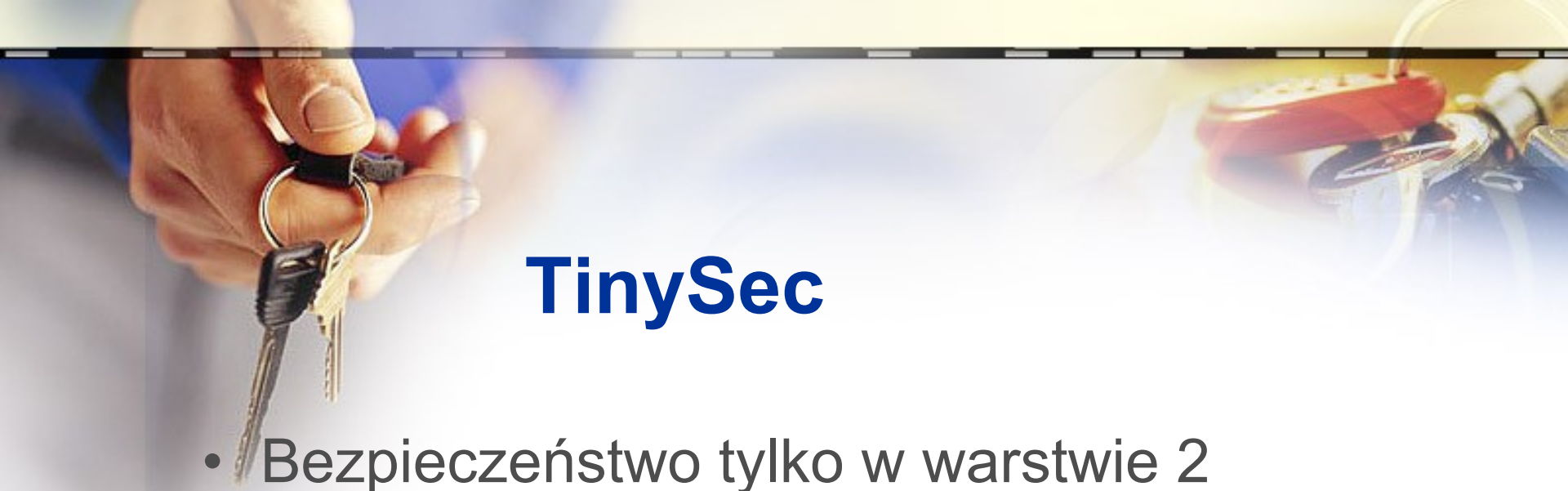


Hello Flood



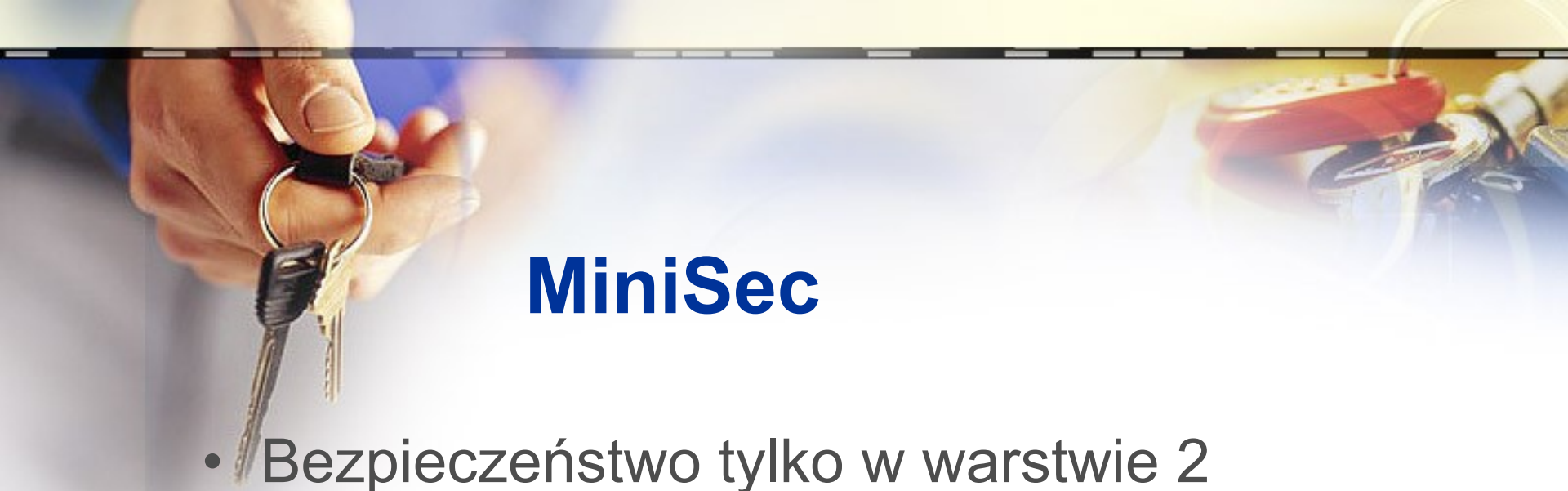
Plan prezentacji

- Sieci sensorowe
- System ROVERS
- Bezpieczeństwo informacji w ROVERS
- Aktualnie istniejące rozwiązania
  - **TinySec**
  - **MiniSec**
- Zaprojektowane rozwiązanie
- Dalsze możliwości rozwoju



TinySec

- Bezpieczeństwo tylko w warstwie 2
- Zapewnienie integralności, uwierzytelnienia i poufności
- Szyfr symetryczny Skipjack w trybach CBC oraz CBC-MAC
- 2 klucze wspólne na całą sieć:
  - **Klucz do szyfrowania**
  - **Klucz do wyliczenia MAC**
- Brak zapewnienia świeżości wiadomości



MiniSec

- Bezpieczeństwo tylko w warstwie 2
- Zapewnienie integralności, uwierzytelnienia, poufności i świeżości danych
- Szyfr symetryczny Skipjack w trybie OCB
- System nic nie mówi o zarządzaniu kluczami
- Tryb OCB jest opatentowany co ogranicza jego użycie

Nowe rozwiązanie

- TinySec i MiniSec to bardziej biblioteki niż pełne rozwiązania
  - **nie spełniają pełnych wymagań stawianych architekturze bezpieczeństwa dla ROVERS**
- Nowe rozwiązanie stworzone na bazie:
  - **trybu GCM**
  - **nowych algorytmów szyfrowania**
- Obejmuje też warstwę aplikacji

Plan prezentacji

- Sieci sensorowe
- System ROVERS
- Bezpieczeństwo informacji w ROVERS
- Aktualnie istniejące rozwiązania
- Zaprojektowane rozwiązanie
  - **Założenia i wymagania**
  - **Algorytmy**
  - **Protokoły**
- Dalsze możliwości rozwoju

Dodatkowe założenia

- W systemie dostępny jest routing, bezstratne dostarczenie wiadomości, fragmentacja
- Jak najmniejszy udział użytkownika w zarządzaniu siecią
- Reprezentant węzła – Mica2
  - **procesor 8 MHz 8-bit Atmel ATMEGA128L**
  - **128 kB pamięci programu**
  - **4 kB pamięci RAM**
  - **512 kB pamięci Flash**
  - **łączność radiowa max. 19.2 kbps**

Wymagania funkcjonalne

- Oddzielenie od siebie komunikacji pochodzącej z różnych sieci
- Bezpieczna komunikacja unicast i broadcast pomiędzy węzłami
- Bezpieczne dodanie/usunięcie urządzenia z sieci
- Bezpieczne przenoszenie kodu i stanu mikro-agentów oraz wprowadzanie nowych aplikacji do działającego systemu

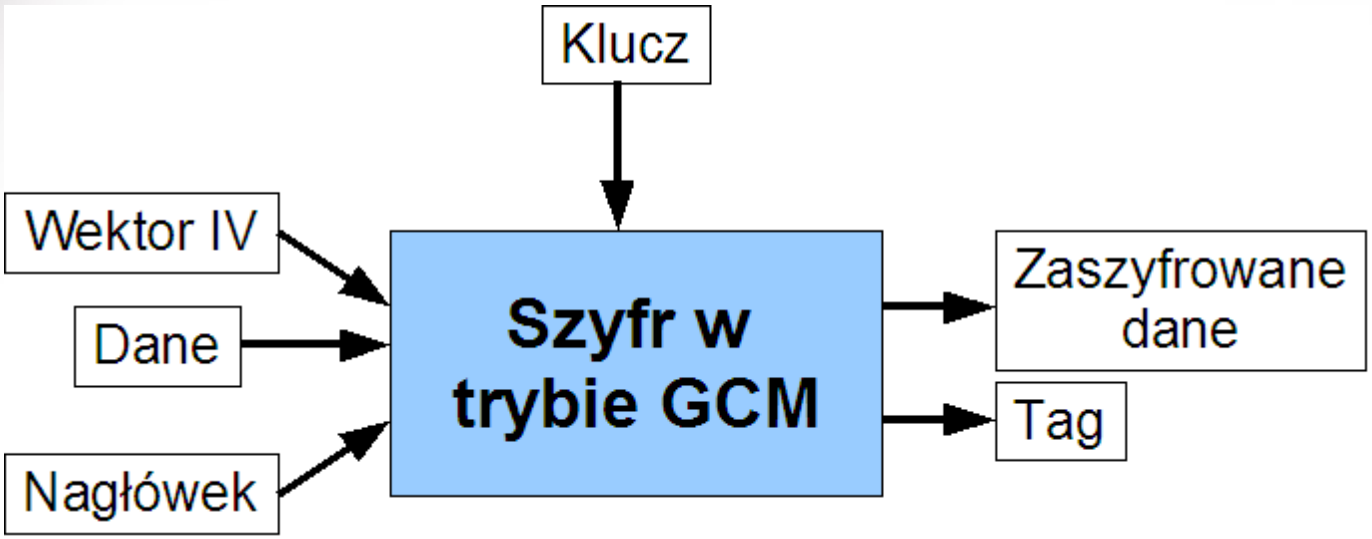
Komunikacja unicast/broadcast (1/2)

- Kryptografia symetryczna
  - **wydajność, oszczędność pamięci**
- Wybrane algorytmy szyfrowania
  - **XTEA (mały kod, brak SBOXów)**
  - **MISTY1 (przeznaczony na mniejsze platformy)**
  - **AES (standard, porównanie)**
- Szyfrowanie w trybie GCM
  - **jednocześnie zapewnienie poufności, integralności i uwierzytelnienia**

Komunikacja unicast/broadcast (2/2)

- Jeden klucz zabezpieczający komunikację radiową dla całej sieci
  - + **Zapewnia elastyczność w ilości węzłów sieci**
  - + **Mniejsza zajętość pamięci**
  - + **Łatwiejsze dodawanie nowych węzłów**
  - - **Zmniejszona odporność na kradzież**
  - - **Wymagana możliwość zmiany klucza na żądanie użytkownika**

Tryb GCM (1/2)

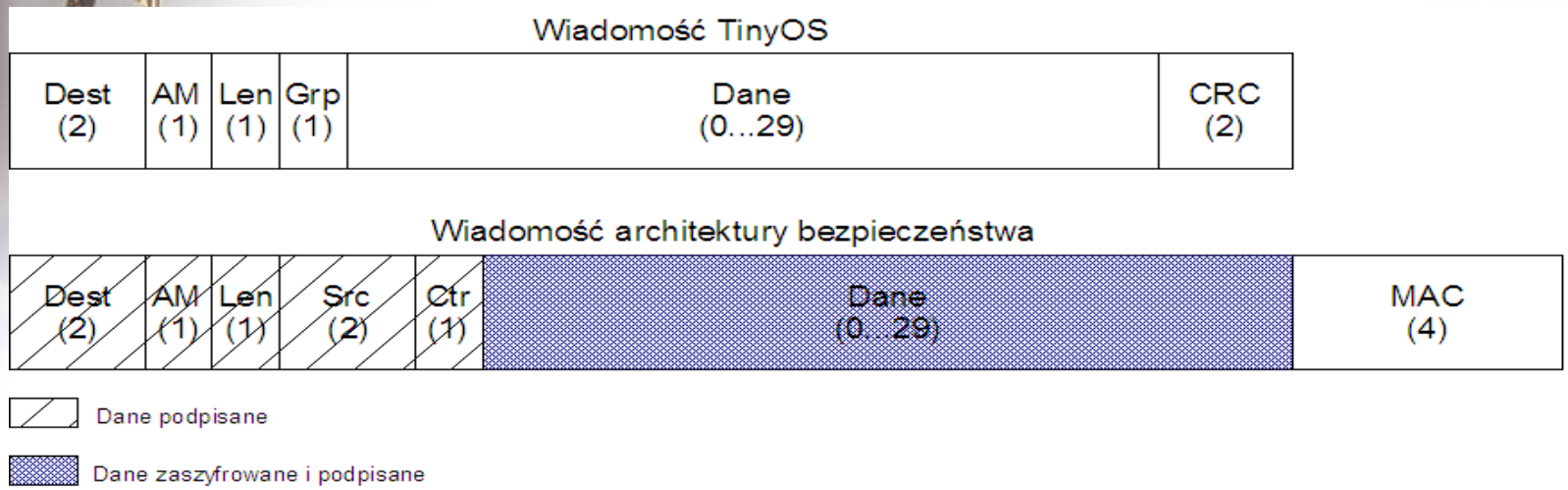


- Zaszyfrowane dane zapewniają poufność
- Tag („podpis”) zapewnia integralność oraz uwierzytelnienie
- Zalecana długość wektora IV – 96 bitów

Tryb GCM (2/2)

- Zalety trybu GCM
  - **Wymaga ceil(PT / BlockSize) + 1 wywołań funkcji szyfrowania**
  - **Wykorzystuje jedynie funkcje szyfrowania**
  - **CT ma taka samą długość co PT**
  - **Zaprojektowany dla szyfrów o wielkości bloku 128 i 64 bitów**
  - **Elastyczny pod względem bezpieczeństwa i wydajności**

Ramka unicast/broadcast



- IV = Dest+AM+Len+Src+Licznik
- Nagłówek - brak
- Ramka o 4 bajty dłuższa niż ramka TinyOS
- Pierwsze 3 bity pola Len użyte do oznaczenia typu wiadomości

Wektor IV

- Musi być unikalny dla każdej ramki, skład:
  - **ID nadawcy**
  - **ID odbiorcy**
  - **Długości wiadomości**
  - **Typu wiadomości**
  - **48 bitowego licznika**
- Licznik zapewnia świeżość i ochronę przed atakami powtórzeniowymi
  - **Odbiorca odrzuca wiadomości z błędnym CTR**
  - **Konieczna synchronizacja liczników**

Dodawanie urządzenia (1/2)

- Wymagana współpraca użytkownika – właściciela sieci
- Użytkownik posiada K*artę sieci* – zawiera ona klucz transmisji oraz klucze zarządzania węzłów
- Karta ma możliwość komunikacji z węzłami
- Dostęp do karty zabezpieczony jest kodem PIN
- W każdym węźle fabrycznie zaszyty jest klucz producenta, który umieszczony jest także np. w instrukcji

Dodawanie węzła (2/2)

1) Użytkownik wpisuje klucz producenta do Karty
2) Karta generuje ID i klucz zarządzania nowego węzła i przesyła te parametry zaszyfrowane kluczem producenta
3) Węzeł odsyła potwierdzenie
4) Karta przesyła klucz transmisyjny bezpiecznym kanałem zarządzania
5) Węzeł posiada wszelkie informacje wymagane do komunikacji. Musi jeszcze zsynchronizować swój licznik broadcast

Usunięcie węzła (1/2)

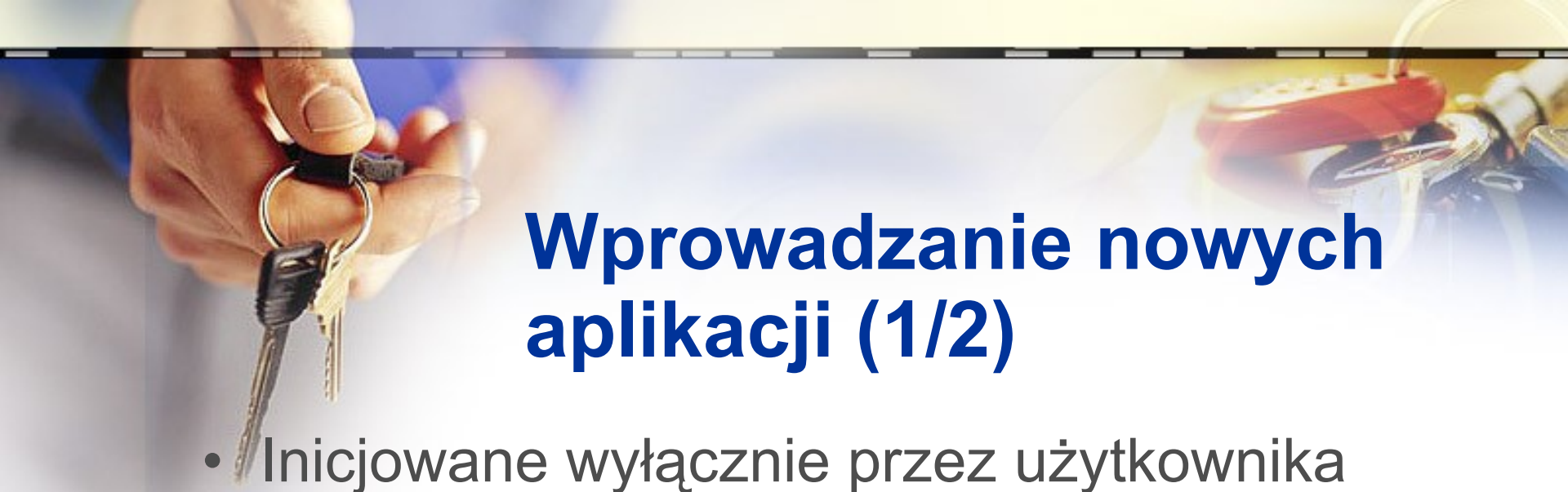
- Musi być wykonane jedynie za zgodą użytkownika
- Musi umożliwiać ponowne dołączenie węzła do tej samej lub innej sieci
- Węzeł nie może zachować kluczy związanych z dawną siecią

Usunięcie węzła (2/2)

1) Użytkownik wybiera węzeł do usunięcia
2) Karta kanałem zarządzania wysyła ramkę usunięcia do węzła
3) Węzeł usuwa z pamięci:
   - **Klucz transmisji**
   - **Stany liczników**
4) Węzeł wysyła potwierdzenie usunięcia i po określonym czasie usuwa klucz zarządzania
5) Karta zaznacza węzeł jako usunięty (usuwa z pamięci licznik, klucz zarządzania tego węzła)

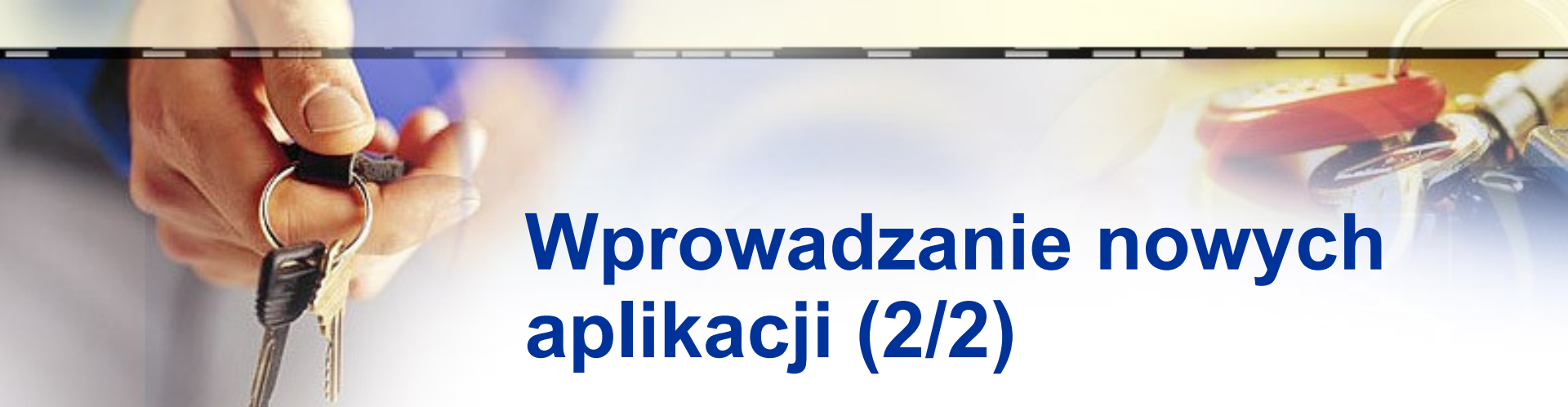
Mobilność mikro-agentów

- Poufność zapewniona przez bezpieczeństwo na poziomie warstwy 2
- Główny wymaganie – zapewnienie, że kod oraz stan mikro-agenta nie uległ zmianie podczas zmiany węzła
- Każdy węzeł przed przekazaniem mikro-agenta oblicza dla niego wartość skrótu SHA1
- Węzeł docelowy akceptuje mikro-agenta tylko w przypadku, gdy wartości skrótu obliczonego i otrzymanego są identyczne



Wprowadzanie nowych aplikacji (1/2)

- Inicjowane wyłącznie przez użytkownika
- Aplikacje znajdują się na *pastylkach aplikacyjnych*
- Istnieje dostęp do pamięci pastylki przed wprowadzeniem jej do sieci
- Wykorzystanie kryptografii asymetrycznej opartej na krzywych eliptycznych (ECC)
- Karta zawiera klucz prywatny sieci, każdy węzeł ma w pamięci klucz publiczny sieci



Wprowadzanie nowych aplikacji (2/2)

1) W czasie zakupu użytkownik podpisuje wybraną aplikację i kopiuje na pastylkę kod transmisyjny oraz przydzielony jej ID
2) Pastylka włącza się do sieci jak zwykły węzeł
3) Po odnalezieniu pierwszego węzła pastylka przesyła do niego aplikację + podpis
4) Jeśli podpis się zgadza węzeł odsyła potwierdzenie i uruchamia aplikację
5) Po uzyskaniu potwierdzenia pastylka jest niepotrzebna i może usunąć swoją zawartość

Plan prezentacji

- Sieci sensorowe
- System ROVERS
- Bezpieczeństwo informacji w ROVERS
- Aktualnie istniejące rozwiązania
- Zaprojektowane rozwiązanie
- Dalsze możliwości rozwoju

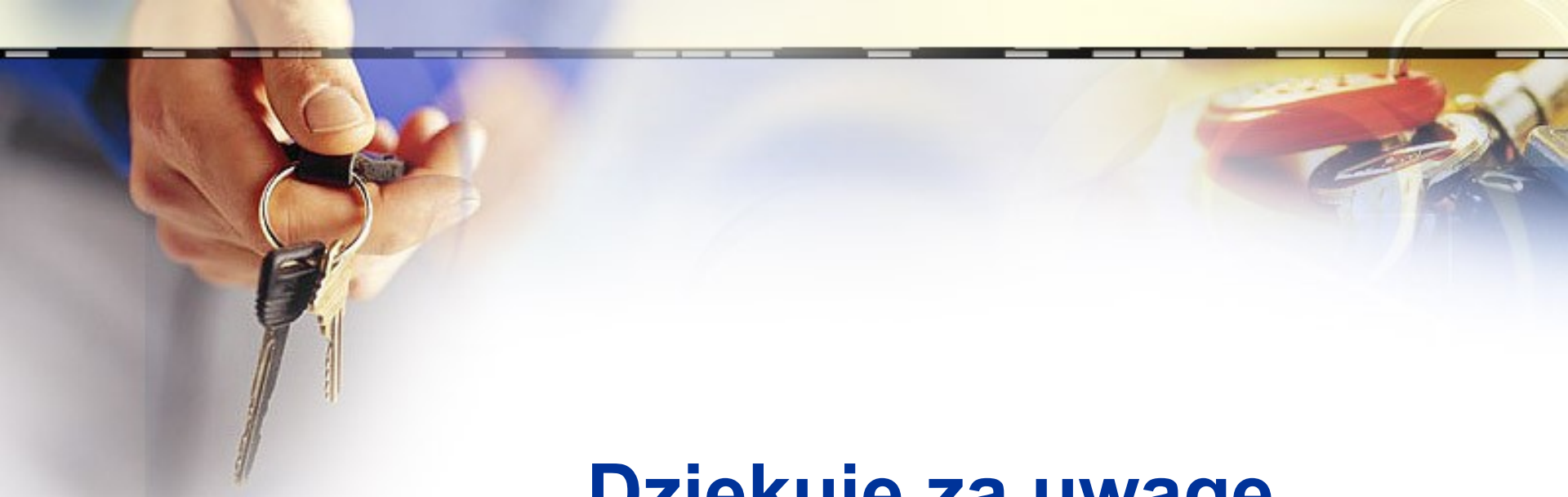
Możliwości dalszego rozwoju

- Integracja z rozwiązaniem zapewniającym synchronizację zegarów w sieci
  - **Usunięcie liczników**
  - **Zwiększenie odporności na ataki powtórzeniowe**
- Szersze wykorzystanie kryptografii asymetrycznej korzystając z silniejszych węzłów (Imote)
  - **Wymaga zmiany reprezentanta węzła**

Intel Mote vs Mica2

- Intel Mote prototyp sensora nowej generacji
- 32-bitowy, 12MHz procesor (8-bit, 8MHz)
- 64 kB pamięci RAM (4 kB)
- 512 kB pamięci Flash (512 kB)
- Radio oparte na standardzie Bluetooth – 2.4 GHz, spread-spectrum, max 720 kbit/s (900 Mhz, jedna częstotliwość, max 19.2 kbit/s)

Na podst. „Intel Mote: An Enhanced Sensor Network Node”, Ralph M. Kling



Dziękuje za uwagę

Pytania ?